



TREBALL FINAL DE GRAU

# MONITORITZACIÓ D'UNA PLANTA DISTRIBUÏDA DE DIPÒSITS INTERCONNECTATS

PROGRAMACIÓ I DISSENY D'UN SCADA

MARTI ANGLADA PAGESPETIT

05/07/2019

GRAU EN ENGINYERIA ELECTRÒNICA INDUSTRIAL I AUTOMÀTICA

EPSEM MANRESA - UPC

## AGRAÏMENTS

El treball final de grau presentat a continuació porta com a títol “ Monitorització d'una planta distribuïda de dipòsits interconnectats”. Aquest treball ha estat escrit com part dels requisits de graduació per el departament d'Enginyeria de la Universitat Politècnica de Catalunya. El període d'investigació i redactat d'aquest treball de fi de grau ha durat des de gener fins a juny de 2019.

El projecte s'ha portat a terme sota la petició i supervisió de la Teresa Escobet Canal. El procés de investigació ha estat llarg i laboriós, però realitzar un estudi exhaustiu m'ha permès resoldre el projecte satisfactòriament. Afortunadament, la meva tutora, la Sra. Teresa Escobet de la EPSEM, sempre ha estat disponible i disposada a ajudar-me amb tots els meus dubtes.

M'agradaria, per tant, donar les gracies a la meva tutora per la seva excel·lent orientació i suport durant tot el procés de realització del meu treball, i sense deixar de banda al servei tècnic de B&R, ja que sense la seva ajuda a l'hora de implementar la comunicació OPC UA, no hagués estat possible portar-la a terme. També m'agradaria donar les gracies a la meva parella Laia per la seva paciència, comprensió i ajuda, ja que sense la seva cooperació no hauria set capaç de portar a terme aquest treball.

També m'ha ajudat discutir sobre varis temes del meu TFG amb els meus amics i família. Si alguna vegada vaig perdre l'interès, vosaltres me la vàreu mantenir motivada. Els meus pares es mereixen també un especial agraïment: els vostres savis consells i recolzament m'han set, com sempre, de gran ajuda.

Espero que gaudiu de la lectura.

Martí Anglada Pagespetit

Manresa, 5 de juliol de 2019.

## RESUM DEL PROJECTE

Aquest projecte de final de grau tracte d'aplicar alguns dels coneixements adquirits al llarg del Grau en Electrónica Industrial i Automàtica, més concretament els adquirits en l'assignatura de Automatització avançada.

Es basa en la necessitat de programar i controlar a través d'una pantalla tàctil una maqueta que simula una planta industrial de dipòsits interconnectats entre ells. La programació se porta a terme mitjançant el llenguatge de programació GRAFCET (Graphe Fonctionel de Commande Etape Transition) i el llenguatge de programació amb text estructurat amb el programa Automation Studio de B&R, i el control i visualització mitjançant un SCADA (Supervisory Control And Data Adquisition) amb el mateix software de B&R, i també per mitjà d'una pantalla tàctil.

Tres autòmats B&R X20CP1381 i un autòmat B&R X20CP1583 faran possible aquesta automatització.

## ABSTRACT

This final project tried to apply some of the knowledge acquired throughout the Degree in Industrial and Automatic Electronics, more specifically those acquired in the course of Advanced Automation.

It is based on the need to program and control through a touch screen, a model that simulates an industrial plant of tanks interconnected among them. The programming is carried out through the programming language GRAFCET (Graphe Fonctionel of Commande Etape Transition) and the programming language with structured text with the B & R Automation Studio program, and the control and visualization through a SCADA (Supervisory Control And Data Adquisition ) with the same B & R software, and also through a touch screen.

Three PLC B&R X20CP1381 and a PLC B&R X20CP1583 will make this automation possible.

## Índex

1.	INTRODUCCIÓ.....	11
1.1	Justificació .....	11
1.2	Antecedents .....	11
1.2.1	Sistemes SCADA .....	11
1.2.1.1	Evolució del SCADA .....	12
1.2.1.2	Com escollir un nou SCADA.....	13
1.2.1.3	Punts clau dels sistemes SCADA.....	14
1.2.1.4	SCADA en la indústria 4.0.....	14
1.2.2	Autòmats programables.....	15
1.2.2.1	Evolució dels autòmats programables .....	16
1.2.2.2	Estandardització .....	17
1.2.2.3	Futur .....	17
1.3	Objectius .....	18
1.4	Plantejament del problema inicial .....	18
2.	COMPONENTS A UTILITZAR .....	19
2.1	Descripció de les maquetes.....	19
2.1.1	Maqueta bàsica .....	19
2.1.1.1	Sensors .....	20
2.1.1.2	Actuadors .....	20
2.1.2	Maqueta distribució .....	21
2.1.2.1	Sensors .....	21
2.1.2.2	Actuadors .....	21
2.1.3	Teclats d'operacions.....	22
2.1.3.1	Teclat d'operacions de les maquetes bàsiques.....	22
2.1.3.2	Teclat d'operacions de la maqueta distribució .....	23
2.2	Descripció dels PLC's i els seus components.....	24
2.2.1	PLC Maquetes bàsiques.....	24
2.2.1.1	Mòdul X1 .....	25
2.2.1.2	Mòdul X2 .....	26
2.2.1.3	Mòdul X3 .....	26
2.2.1.4	Mòdul X4 .....	27
2.2.1.5	Assignació d'entrades i sortides.....	28

2.2.2	PLC Maqueta distribució .....	29
2.2.2.1	Mòdul d'alimentació i comunicacions.....	30
2.2.2.2	Mòdul X20DIF371 .....	30
2.2.2.3	Mòdul X20DOF322 .....	31
2.2.2.4	Assignació d'entrades i sortides .....	32
2.3	Pantalla tàctil.....	33
3.	ENTORN DE PROGRAMACIÓ .....	34
3.1	B&R Automation Studio .....	34
3.1.1	Creació d'un nou programa.....	34
3.1.2	Configuració d'un sistema de visualització .....	40
3.1.2.1	Actualització de la imatge i el hardware de la pantalla .....	40
3.1.2.2	Afegir i configurar el terminal al projecte de Automation Studio.....	41
3.1.2.3	Disseny de la pantalla.....	43
3.1.3	Sistema de comunicació OPC UA .....	47
3.1.3.1	Definició de OPC UA .....	47
3.1.3.2	Activació Sistema OPC UA .....	47
3.1.3.3	Configuració Sistema OPC UA .....	49
3.1.3.4	Servidor OPC UA.....	49
3.1.3.5	Client OPC UA.....	50
4.	METODOLOGIA.....	53
4.1	Programació maquetes base.....	53
4.1.1	Diagrama GEMMA maquetes base .....	53
4.1.1.1	Grafcet diagrama GEMMA maquetes base.....	55
4.1.1.2	Grafcet parada d'emergència maquetes base .....	56
4.1.2	Espera de recepta maquetes base .....	57
4.1.3	Recepta 1 maquetes base .....	57
4.1.4	Recepta 2 maquetes base .....	58
4.1.5	Recepta 3 maquetes base .....	59
4.2	Programació maqueta distribució.....	60
4.2.1	Diagrama GEMMA maqueta distribució .....	61
4.2.1.1	Grafcet diagrama GEMMA maqueta distribució.....	61
4.2.1.2	Grafcet parada d'emergència maqueta distribució .....	62
4.2.2	Espera de recepta maqueta distribució .....	64
4.2.3	Recepta 4 maqueta distribució .....	64
4.2.4	Recepta 5 maqueta distribució .....	66

4.2.5	Recepta 6 maqueta distribució .....	66
4.3	Pantalles de visualització.....	68
4.3.1	Pantalla principal.....	68
4.3.2	Pantalla de receptes.....	69
4.3.3	Pantalla de visualització maquetes 1, 2 i 3.....	70
4.3.4	Pantalla de visualització maqueta de distribució.....	73
4.3.5	Pantalla de gràfica de nivell i control PID.....	74
5.	POSSIBILITATS DE MILLORA .....	75
6.	CONCLUSIONS .....	76
7.	REFERÈNCIES .....	77
8.	ANNEXES .....	79
8.1	Programes i variables maquetes base 1, 2 i 3 .....	79
8.1.1	Variables globals.....	79
8.1.2	Programa Grafcets 100 i 120.....	80
8.1.3	Programa Espera de recepta .....	82
8.1.4	Programa Recepta 1 .....	84
8.1.5	Programa Recepta 2 .....	85
8.1.6	Programa Recepta 3 .....	88
8.2	Programes i variables maqueta distribució.....	90
8.2.1	Variables globals.....	90
8.2.2	Programa Grafcets 200 i 220.....	93
8.2.3	Programa Espera de recepta .....	95
8.2.4	Programa de Receptes .....	97
8.2.5	Programa Recepta 4 .....	98
8.2.6	Programa Recepta 5 .....	100
8.2.7	Programa Recepta 6 .....	102
8.2.8	Programa control de visualització.....	104

## Índex d'imatges

Imatge 1: Exemple de SCADA industrial [4] .....	11
Imatge 2: Sala de control amb sistema SCADA [4] .....	12
Imatge 3: Exemple de disseny d'un sistema de monitorització [6] .....	13
Imatge 4: Piràmide control i gestió industrial [9] .....	15
Imatge 5: Línia del temps, evolució autòmats programables [14] .....	16
Imatge 6: Esquema maquetes bàsiques .....	19
Imatge 7: Esquema maqueta distribució .....	21
Imatge 8: Teclat operacions maqueta 1 .....	22
Imatge 9: Teclat operacions maqueta distribució .....	23
Imatge 10: PLC B&R X20CP1381 .....	24
Imatge 11: Característiques PLC X20CP1381 .....	24
Imatge 12: Mòdul E/S X1 .....	25
Imatge 13: Mòdul E/S X2 .....	26
Imatge 14: Mòdul E/S X3 .....	27
Imatge 15: Mòdul sortides analògiques X4 .....	27
Imatge 16: PLC B&R X20CP1583 .....	29
Imatge 17: Característiques PLC X20CP1583 .....	29
Imatge 18: Mòdul alimentació i comunicacions .....	30
Imatge 19: Mòdul X20DIF371 .....	30
Imatge 20: Mòdul X20DOF322 .....	31
Imatge 21: Pantalla HMI .....	33
Imatge 22: Característiques de la pantalla HMI .....	33
Imatge 23: Versió utilitzada Automation Studio .....	34
Imatge 24: Nou Projecte AS .....	34
Imatge 25: Nom i ubicació nou projecte AS .....	35
Imatge 26: Escollir configuració hardware AS .....	35
Imatge 27: Entorn de treball del AS .....	36
Imatge 28: Declaració variables globals .....	37
Imatge 29: Afegir mòdul de programa .....	37
Imatge 30: Assignació variables a E/S .....	38
Imatge 31: Configuració mòdul Ethernet .....	39
Imatge 32: Assignació programes al cicles de treball .....	39
Imatge 33: Actualització del programari de AS .....	40
Imatge 34: Pantalla d'actualització d'AS .....	41
Imatge 35: Afegir hardware pantalla tàctil .....	41
Imatge 36: Configuració Startup pantalla .....	42
Imatge 37: Assignar arxiu d'imatge a la pantalla .....	42
Imatge 38: Configuració IP VNC Server pantalla .....	42
Imatge 39: Afegir element de visualització .....	43
Imatge 40: Espai de treball, disseny de visualitzacions .....	43
Imatge 41: Objectes visualització .....	44
Imatge 42: Elements visualització .....	44
Imatge 43: Propietats dels objectes de visualització .....	45



Imatge 44: Afegir un botó a una pagina de visualització .....	45
Imatge 45: Configuració botó en una visualització .....	46
Imatge 46: Diagrama de comunicació OPC UA .....	47
Imatge 47: Activar Sistema OPC UA .....	48
Imatge 48: Configuració propietats sistema OPC UA.....	48
Imatge 49: Arxiu de configuració OPC UA.....	49
Imatge 50: Afegir hardware configuració OPCUa_any .....	50
Imatge 51: Configuració dispositiu OPCUa_any.....	51
Imatge 52: Canals d'entrada i sortida OPC UA.....	51
Imatge 53: Assignació de variables al canal OPC UA.....	52
Imatge 54: Diagrama GEMMA maquetes base .....	54
Imatge 55: Grafcet 100, Diagrama GEMMA maquetes base .....	55
Imatge 56: Grafcet 120, parada d'emergència maquetes base .....	56
Imatge 57: Grafcet 0, espera de recepta maquetes base .....	57
Imatge 58: Grafcet 10, Recepta 1 maquetes base .....	58
Imatge 59: Grafcet 20, recepta 2 maquetes base.....	59
Imatge 60: Grafcet 30, recepta 3 maquetes base.....	60
Imatge 61: Diagrama GEMMA maqueta distribució .....	61
Imatge 62: Grafcet 200, diagrama GEMMA maqueta distribució.....	62
Imatge 63: Grafcet 220, parada d'emergència maqueta distribució .....	63
Imatge 64: Grafcet 40, espera de recepta maqueta distribució.....	64
Imatge 65: Grafcet 50, recepta 4 maqueta distribució .....	65
Imatge 66: Grafcet 60, recepta 5 maqueta distribució .....	66
Imatge 67: Grafcet 70, recepta 6 maqueta distribució .....	67
Imatge 68: Pantalla principal SCADA.....	68
Imatge 69: Pantalla receptes SCADA.....	69
Imatge 70: Pantalla control i visualització maqueta 1 .....	70
Imatge 71: Panell de control maqueta 1 .....	70
Imatge 72: Panell de control principal maqueta 1 .....	71
Imatge 73: Control i visualització esquematitzat maqueta 1.....	72
Imatge 74: Pantalla control i visualització maqueta distribució .....	73
Imatge 75: Pantalla de visualització de la gràfica de nivell .....	74

## Índex de taules

Taula 1: Glossari .....	10
Taula 2: Assignació Entrades PLC X20CP1381 .....	28
Taula 3: Assignació Sortides PLC X20CP1381 .....	28
Taula 4: Assignació Entrades PLC X20CP1583 .....	32
Taula 5: Assignació Sortides PLC X20CP1583 .....	32

## Glossari

Taula 1: Glossari

Paraula	Descripció
GRAFCET	Model de representació gràfica o diagrama funcional normalitzat, que permet fer un model del procés a automatitzar, contemplant entrades, accions a realitzar, i els processos intermedis que provoquen aquestes accions.
HMI	Interfície d'usuari per les seves sigles en anglès, (Human (I) Machine Interface) que s'usa per referir-se a la interacció entre humans i màquines; Aplicable a sistemes d'Automatització de processos.
MES	Sistemes computats utilitzats en la fabricació per rastrejar i documentar la transformació de matèries primeres en productes acabats.
PLC	En anglès (Programmable Logic Controller) és un autòmat industrial dissenyat per controlar processos seqüencials, és un dispositiu electrònic que és pot programar per l'usuari.
SCADA	En anglès (Supervisory Control And Data Acquisition). Es tracta d'una aplicació software especialment dissenyada per funcionar en ordinadors en el control de la producció, proporcionant comunicació amb els dispositius de camp.
SERVIDOR OPC	Aplicació de programari (driver) que compleix amb una o més especificacions definides per l'OPC Foundation. El Servidor OPC fa d'interfície comunicant d'una banda amb una o més fonts de dades utilitzant les seves protocol nadius (típicament PLC's) i per l'altre costat amb Clients OPC (típicament SCADAS , HMIS, etc.)
OPC UA	OPEN Connectivity Unified Architecture (OPC UA) és un estàndard internacional per a un intercanvi d'informació segur, fiable, independent del fabricant i de la plataforma en la comunicació industrial.

## 1. INTRODUCCIÓ

El següent projecte es basa en, utilitzant les 4 estacions funcionals del fabricant FESTO equipades amb diversos dipòsits, sensors i actuadors, crear un programa utilitzant els autòmats programables de la marca *B&R Industrial Automation* de la sèrie X20 i dissenyar una visualització i control per poder interactuar amb les maquetes a través d'una pantalla tàctil del mateix fabricant que els PLC's.

En el projecte a mes a mes, al haver d'interactuar amb diferents autòmats i davant la necessitat de compartir senyals i variables entre ells, neix la obligació d'incorporar un servidor OPC per aconseguir-ho.

Paral·lelament a l'esmentat anteriorment es vol familiaritzar-se amb un entorn de control utilitzant el software de programació de *B&R l'Automation Studio* i adquirir nous coneixements sobre automatització i control de PLC's industrials i el disseny i programació de sistemes SCADA.

Per a la realització d'aquest treball fi de grau s'han seguit les directrius descrites a [1].

### 1.1 Justificació

L'elecció d'aquest tema per el TFG ve donada per la inquietud i necessitat adquirida al llarg d'aquests anys de formació i aprenentatge, per poder automatitzar, monitoritzar i controlar un procés el mes semblant i pròxim possible a la realitat industrial actual.

### 1.2 Antecedents

#### 1.2.1 Sistemes SCADA

A la dècada dels anys 60, els sistemes de Supervisió, Control y Adquisició de Dades, més coneguts com "SCADA" (del anglès, "Supervisory Control and Data Acquisition"), van començar a popularitzar-se a les economies industrialitzades per fer front a les creixents necessitats de monitoritzar i controlar equipament remot, tant en el sector industrial com en les 'Utilities'. Amb el pas del temps i la conseqüent evolució de la tecnologia, aquests sistemes han passat de ser 'mainframes' monolítics a fer servir arquitectures mes flexibles i que els hi han permès incorporar una major seguretat i precisió, així com també nivells més alts de funcionalitat.



Imatge 1: Exemple de SCADA industrial [4]

Entre les aplicacions d'una plataforma SCADA, es troben els següents elements [1]:

- Centralitzar informació.
- Controlar a distància.
- Conèixer tendències i històrics del sistema.
- Manipular alarmes.
- Realitzar anàlisis predictiu i preventiu d'una planta industrial.

Els sistemes SCADA han estat una part fonamental en els processos industrials dels últims temps. Però, ¿en què consisteixen? Un sistema SCADA s'encarrega de visualitzar, gestionar i supervisar processos, seqüències de fabricació de màquines i plantes en tots els sectors disponibles. Abasta des de sistemes simples fins a solucions diversificades o complexes com la indústria de l'automòbil i, a més, constitueixen una plataforma d'intercanvi d'informació per a la integració vertical en l'àmbit corporatiu.

#### 1.2.1.1 Evolució del SCADA

De la mà dels avenços en les Tecnologies de la Informació (TI), els fabricants també han anat implementant noves prestacions en els seus productes. Entre elles, Gabriel Castro, Product Manager, Factory Automation Systems de Siemens ressalta la introducció d'estàndards TI "moderns i pràctics", com SQL i les interfícies web. "Això ha millorat enormement l'eficàcia, la seguretat, la productivitat i la fiabilitat d'aquestes plataformes", indica.

En la mateixa línia, Diego Valencia, Cap de Suport Tècnic Comercial de RHONA, comenta que els SCADA han anat avançant cap a una major integració amb tots els processos informàtics de l'empresa. "Les connexions amb ERP i bases de dades empresarials ja no són estranyes, tenint informació fiable per alimentar els sistemes de gestió gerencials. A més, la capacitat actual de guardar informació històrica a taxes de mostreig de pocs mil·lisegons permet descobrir tendències abans desconegudes", explica.



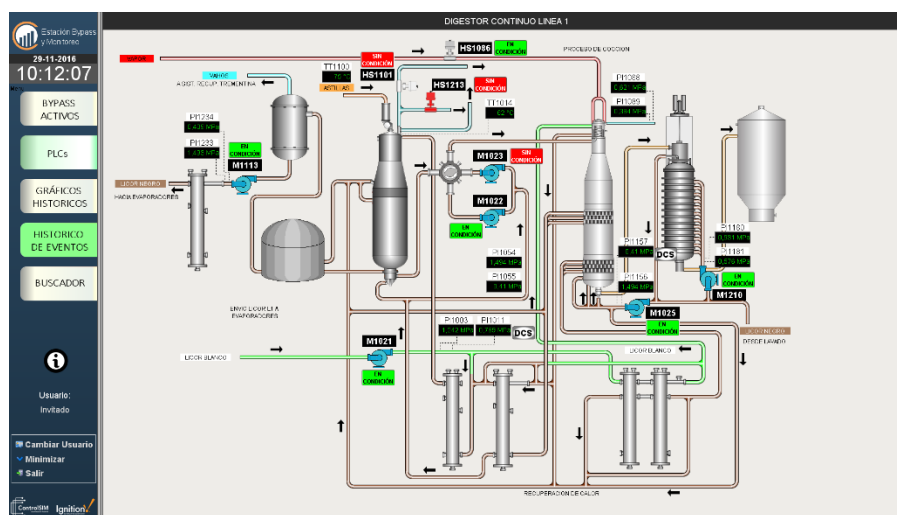
Imatge 2: Sala de control amb sistema SCADA [4]

Coincidint amb l'opinió anterior, Robert Bauerschmitt, Gerent de Producte de Fabelec, emfatitza els avenços en la interconnexió amb altres sistemes informàtics. "Ara ve la Inter connectivitat amb l'anomenada 'Internet de les Coses', permetent alimentar les solucions de Big Data", comenta.

Per la seva banda, Bastián Botxí, Enginyer d'Aplicacions i Vendes en Tecnet Xile, destaca les diferents funcionalitats que s'han anat incorporant a aquestes plataformes. "Per exemple, s'han desenvolupat diversos mòduls de programari que permeten implementar models que, alimentats amb la informació del SCADA, generen respostes que ajuden a l'operació en línia com també a provar escenaris simulats i veure els seus efectes en el procés", sosté.

### 1.2.1.2 Com escollir un nou SCADA

Donat el paper que juguen aquestes plataformes dins d'un procés productiu, triar el sistema SCADA adequat pot ser summament complex, tenint en compte la gran gamma de fabricants, funcionalitats i opcions existents en el mercat. Per exemple, ¿quin abast ha de tenir la marca en l'elecció, donades les actuals capacitats de comunicació dels equips i sistemes a interconnectar?



Imatge 3: Exemple de disseny d'un sistema de monitorització [6]

Per triar la plataforma adequada, es recomana definir un sistema SCADA acord amb el procés que es vol controlar, més enllà del cost. "Per a una aplicació en distribució d'energia, no és el mateix seleccionar un sistema per a propòsits generals que un sistema específic per a aquesta aplicació. En cas del sistema de propòsits generals podria tenir un cost inicial menor, però eventualment podria no comptar amb les eines específiques requerides per a un procés de distribució d'energia. L'últim, provocaria complexitats en la configuració del sistema i manteniment en veure forçat a emprar eines adaptades per a complir una funció per a les que no van ser desenvolupades en forma nativa", comenta Bastián Verdugo, Enginyer d'Aplicacions i Vendes en Tecnet Chile. "Les tecnologies van convergint a passos de gegant, de manera que un sistema SCADA ha de tenir la capacitat d'anar evolucionant en conjunt amb aquests canvis (nous equips, actualització de protocols, augment de variables, canvi de sistemes operatius, etc.). Si un sistema SCADA no té la capacitat d'avançar ràpid amb aquests canvis l'actualització és molt costosa, per la qual cosa és moment de prendre la decisió de renovar un sistema", sosté Claudio Gajardo, Supervisor de Vendes de Sistemes de Control de Yokogawa Chile.

### 1.2.1.3 Punts clau dels sistemes SCADA

- **FUNCIONS HMI**

Qualsevol SCADA no ha d'obviar les funcions d'interfície Home-Màquina (HMI), és a dir, maneig i visualització del procés, funcions d'alarma i justificant d'esdeveniments, històric de valors per a la seva posterior anàlisi, documentació i informes de producció.

- **ESTÀNDARDS OBERTS**

Avui dia es valora en gran mesura que la comunicació amb sistemes tercers no sigui propietària, es poden incorporar clients estàndard OPC UA per tal d'intercanviar dades fàcilment amb sistemes tipus MES (Manufacturing Execution System) o amb sistemes SAP, fins i tot amb una interfície basada en arxius XML. Interactuar amb el món IT serà part fonamental de les fàbriques d'avui i de demà [3].

- **ESCALABILITAT**

Les arquitectures típiques d'un SCADA van des d'un simple sistema amb una màquina individual en la qual s'aglutinen tots els serveis HMI fins a sistemes distribuïts amb diferents llocs en els quals la informació està diversificada entre diversos llocs i tots ells comuniquen entre si. Sense oblidar-nos l'alta disponibilitat en planta, que es cobreix amb sistemes redundants capaços de balancejar la càrrega d'un equip a un altre davant de qualsevol anomalia.

- **IoT (INTERNET OF THINGS)**

És un concepte que es refereix a la interconnexió digital d'objectes quotidians amb internet. Si ens centrem en la indústria, podem veure com tots els sistemes de producció industrial ja no poden estar aïllats, han de interactuar entre ells i d'aquesta manera reduir els temps de producció i que el 'time to market' sigui menor.

- **CIBERSEGURETAT**

La ciberseguretat cobreix un ventall molt ampli de possibilitats de protecció a nivell de protecció d'accés, seguretat IT i, finalment, seguretat de planta. És a dir, hem de tenir en compte des de la prevenció física a persones no autoritzades, passant per una protecció en les xarxes IT de planta, fins a finalment autenticant usuaris i operaris de màquina.

### 1.2.1.4 SCADA en la indústria 4.0

Avui en dia, el concepte d'Indústria 4.0, també assenyalat com a indústria intel·ligent o indústria del futur, correspon amb una nova manera d'entendre els mitjans de producció [2]. Aquest nou fenomen no és una realitat ja consolidada i experimentada, sinó una fita en el desenvolupament industrial que sens dubte marcarà els processos productius en els propers anys. Tenint en compte nous conceptes com la digitalització, la coordinació corporativa,

l'Internet de les Coses són i seran els punts en els quals el SCADA ha de donar resposta a una indústria ja consolidada i experimentada.

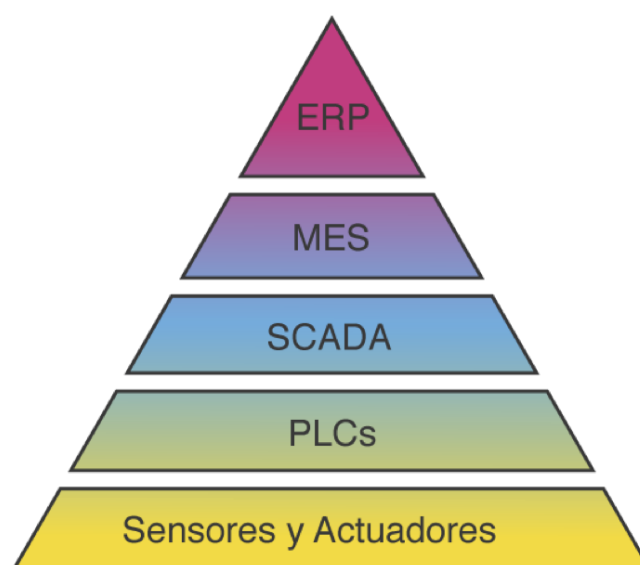
### 1.2.2 Autòmats programables

La tecnologia dels autòmats programables porta anys aplicant-se a la indústria de la mà de l'enginyeria, especialment des que prengué més força en els anys 60, això es deu primordialment a la gran adaptabilitat, versatilitat i senzillesa en la seva implementació. Els processos automatitzats ofereixen nombrosos avantatges enfront el control humà de les màquines, ja que suprimeix l'operari de realitzar múltiples tasques repetitives, avorrides o perilloses, augmentant l'eficiència dels processos notablement. per tant es pot dir que la introducció dels autòmats a la indústria ha proporcionat un control més eficient, segur i ràpid dels processos industrials [5].

Pel que podem dir que avui en dia els autòmats programables són un element imprescindible en l'automatització. Malgrat tot, els autòmats programables es poden utilitzar en molts altres camps a part de la indústria, com poden ser el transport, la medicina, la domòtica, etc.

Tradicionalment, un sol autòmat controlava tot el procés, el que es denomina control centralitzat, però ara gràcies a les xarxes industrials podem parlar de control distribuït. Aquest tipus de xarxes industrials han estat necessàries per controlar grans processos industrials, així diversos autòmats s'encarreguen del control total del procés, però cada un d'ells controla una part del mateix.

L'autòmat és l'encarregat de gestionar la informació que li arriba del procés per prendre les accions de control necessàries.



*Imatge 4: Piràmide control i gestió industrial [9]*



### 1.2.2.1 Evolució dels autòmats programables

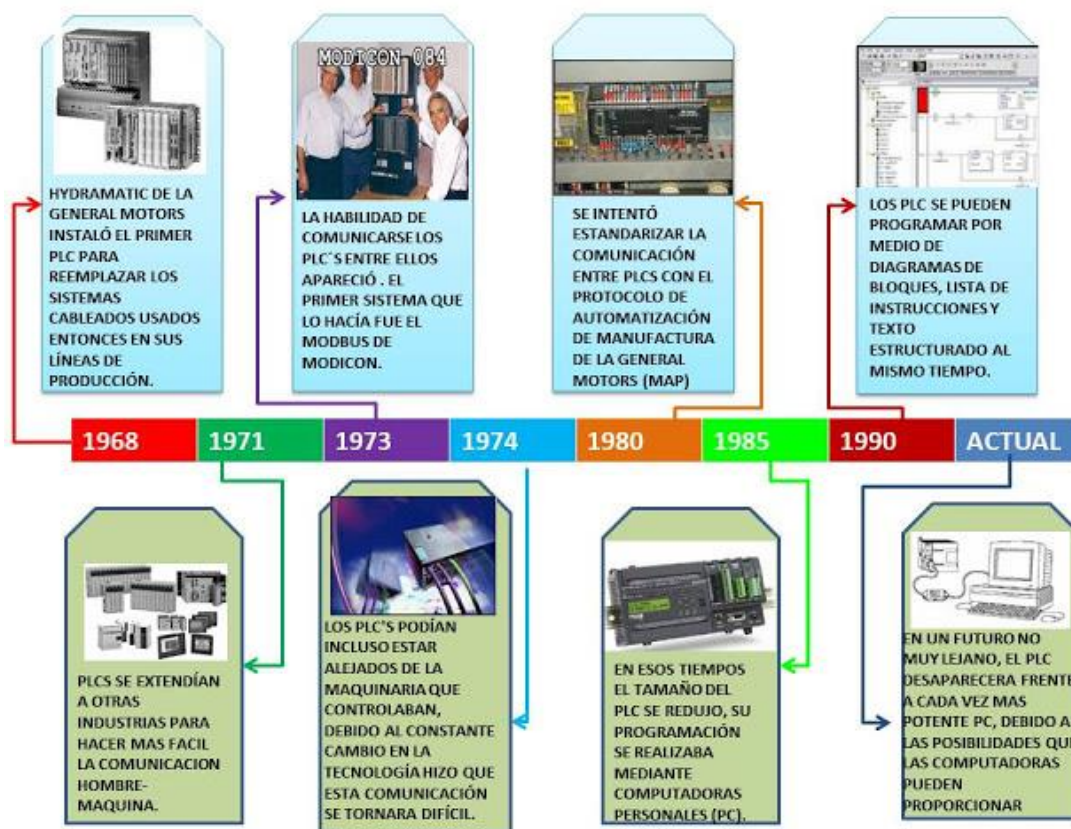
Durant aquests més de 40 anys, els PLC's han evolucionat en moltíssims aspectes i, si en els seus inicis la seva missió fonamental era substituir a la lògica cablejada, avui dia, amb els avenços tecnològics i la gran reducció de costos, el seu objectiu arriba molt més enllà que el de fer càlculs lògics.

Així, a més del control de la lògica discreta, el control de processos continus amb senyals analògics o de sondes de temperatura va ser una de les primeres incorporacions al món del PLC.

Però ha estat a partir de l'any 2000 quan el món del PLC ha patit un major avanç. Amb la reducció de costos en els equips electrònics més complexos i en l'electrònica de potència, van sorgir els controladors específics per a visió artificial, per 'Motion Control', seguretat, etc.

Amb tots aquests controladors, a part del PLC, va sorgir la necessitat imperiosa de la integració: amb diferents xarxes de comunicació, diferent programari de configuració, diferents llenguatges de programació, etc.

Avui dia, en l'afany de simplificar la integració de tots aquests sistemes, sorgeix el concepte de MAC (Machine Automation Controller) o PAC (Programmable Automation Controller), on el PLC evoluciona per suportar moltes de les disciplines abans esmentades. De manera que des d'un únic programari, amb un llenguatge i interfície comú i sense cap problema d'interconnexió, un únic controlador pugui fer-se càrrec de l'automatització d'una màquina o una línia de producció sencera.



Imatge 5: Línia del temps, evolució autòmats programables [14]

### 1.2.2.2 Estandardització

Un dels problemes que va aparèixer amb l'evolució i comercialització dels diferents models de PLC, és la fragmentació en els llenguatges de programació usats en cada un dels equips, accentuat més per llenguatges propis de les diferents marques.

El llenguatge amb el qual va començar tot va ser el 'ladder'. Aquest llenguatge, es va inventar per representar d'una manera clara la lògica realitzada amb relés, i el diagrama, amb les diferents connexions i contactes, recorda al d'una escala ( 'ladder' en anglès) amb les seves diferents esglaons ( 'rungs'). No obstant això, amb la inclusió de nova funcionalitat en els PLCs van anar apareixent nous llenguatges, més adequats per realitzar altres tasques, com càlculs aritmètics complexos, sentències condicionals o que permetien una major reutilització del codi.

Així, a finals de 1993, la CEI (Comissió Electrotècnica Internacional) va publicar la primera revisió de l'estàndard 61131, que en el seu apartat tercer defineix els llenguatges de programació estàndard d'un PLC. Actualment són els següents:

- Ladder Diagram (LD, diafragma d'escala)
- Function Block Diagram (FBD, diagrama de blocs de funció)
- Structured Text (ST, text estructurat)
- Instruction List (IL, llista d'instruccions)
- Sequential Function Chart (SFC, gràfic de funcions seqüencials)

Això proporciona un ventall molt important de possibilitats en la programació dels controladors actuals. Fins i tot, alguns fabricants permeten barrejar diversos d'ells en un mateix programa, fent que la tasca de desenvolupament sigui molt més senzilla.

D'altra banda, la norma IEC anterior, no només es limita a definir els llenguatges de programació, sinó també els tipus de dades i alguns aspectes propis del funcionament del propi controlador: gestió de dades usant variables, ús de tasques, etc.

Aquesta estandardització, al costat d'altres, fa que d'una manera molt senzilla, es puguin utilitzar diversos controladors sense haver d'aprendre cap nou llenguatge o forma de programar.

### 1.2.2.3 Futur

La integració de la part de PLC, de 'Motion Control', de visió, de robòtica, de 'Safety', de CNC, etc., no només porta amb si una homogeneïtat a l'hora de desenvolupar sinó que permet una integració total entre cada una de les disciplines a implementar permetent compartir variables, dades i una sincronització i precisió absoluta entre cada un dels elements de la màquina.

Ahora, el programari evoluciona creant veritables estudis de desenvolupament, des d'on es pot configurar, ja no només el controlador, si no cadascuna de les diferents parts de la màquina, des d'un interfície comú.

Aquest mateix programari ha de ser l'encarregat de guiar-nos durant tot el procés de desenvolupament, assistint-nos i ajudant-nos en cada un dels aspectes de configuració i programació. Però més enllà de permetre'ns programar un dispositiu concret, serà capaç de permetre'ns i assistir-nos en la creació d'una màquina completa.

Un pas encara més enllà, serà la comprovació i el test del nostre desenvolupament. Les millors eines actuals ja permeten fer una simulació de totes les disciplines implementades en la màquina (seqüència, control de moviment, visualització, etc.) i detectar i corregir errors abans fins i tot de tenir el maquinari. Però no només això, sinó que el disseny en 3D de la màquina i la integració amb altres eines de regulació i processos, com Matlab permetran des de l'eina de programari dissenyar tota la màquina i assegurar, fins i tot abans de muntar-la, que funciona.

No hi ha dubte que el futur ens depara controladors més ràpids, potents, amb més memòria, etc., però al mateix temps amb funcions útils, que facin més fiables les màquines i que permetin un desenvolupament molt més ràpid.

### 1.3 Objectius

Aquest treball de fi de grau té per objectiu la integració de les quatre maquetes disponibles, programar cada un dels autòmats del fabricant B&R per automatitzar un procés el més semblant possible a una procés industrial de productes líquids on tindrem diferents receptes per fabricar el producte que desitgem i paral·lelament dissenyar i desenvolupar un sistema de visualització i control SCADA utilitzant una pantalla tàctil del mateix fabricant per poder monitoritzar totes les senyals i controlar totes les maquetes poden així engegar i parar els processos, i modificar les receptes.

L'objectiu principal està compost pels següents objectius específics:

- Dissenyar i elaborar el programa per cada un dels PLC's a partir de GRAFCETS.
- Integrar els 4 autòmats programables utilitzant el protocol de comunicació OPC UA.
- Dissenyar i desenvolupar un 'SCADA' per monitoritzar i controlar tot el procés .

### 1.4 Plantejament del problema inicial

El projecte s'ha de realitzar a l'edifici de l'EPSEM, en el departament d'automatització de la universitat, concretament a l'aula laboratori d'automatització.

Es demana automatitzar, dissenyar i controlar via pantalla tàctil una maqueta de laboratori que simula una planta industrial de dipòsits interconnectats, fent ús dels sensors i actuadors que aquesta maqueta posseeix.

Encara que la maqueta disposa de botonera, el control s'ha de realitzar paral·lelament mitjançant un SCADA utilitzant una pantalla tàctil, de manera que el procés es pugui posar en marxa, parar-se i també visualitzar-se des de l'SCADA de la pantalla tàctil.

## 2. COMPONENTS A UTILITZAR

En aquest punt es descriuran tots els components que s'utilitzaran per a la realització del projecte, tant els seus components com característiques i funcionament dels mateixos.

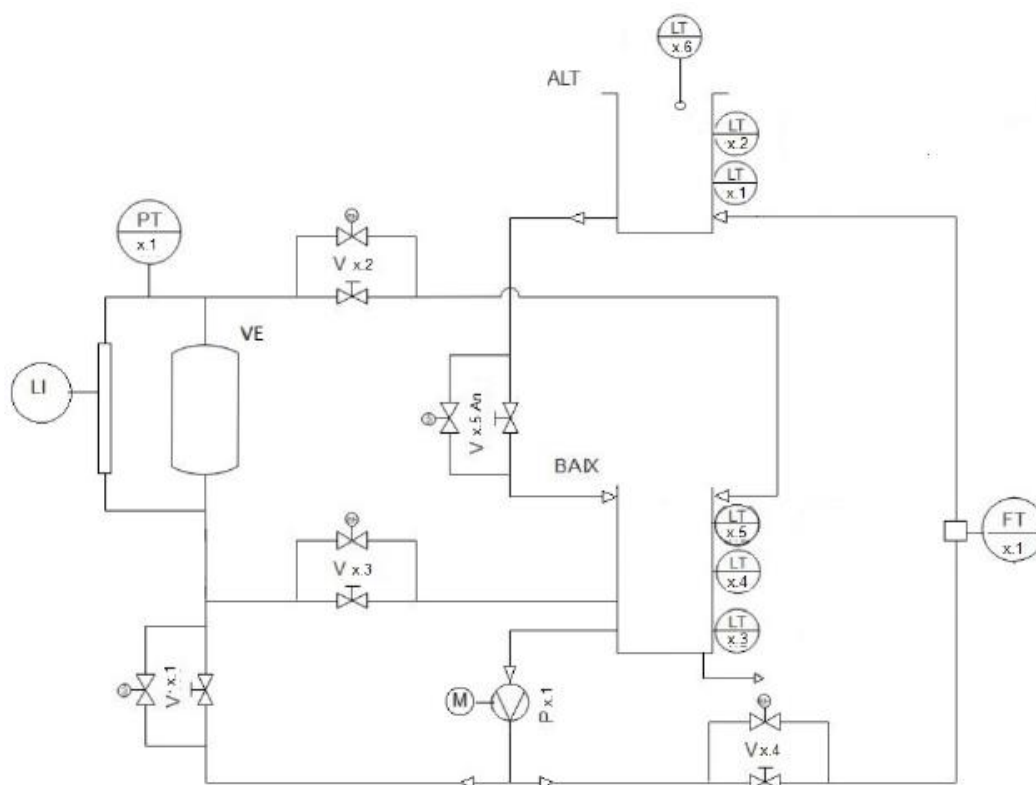
### 2.1 Descripció de les maquetes

#### 2.1.1 Maqueta bàsica

Es disposa de tres estacions de treball semblants, numerades del 1 al 3. Cada una d'elles consta de 2 dipòsits ubicats a diferents alçades i un vas de pressurització, tots ells interconnectats mitjançant tubs.

Aquestes maquetes estan formades per 8 sensors com a entrades i 6 actuadors com a sortides.

L'esquema complert de la maqueta es mostra a la Imatge 6, la x té els valors de 1, 2 o 3 en funció de la maqueta de treball.



Imatge 6: Esquema maquetes bàsiques

Com és pot observar a la Imatge 6 l'aigua es pot portar del tanc BAIX al tanc ALT activant la motobomba Px.1 i obrint electrovàlvula Vx.4, i es pot retornar al tanc BAIX obrint l'electrovàlvula Vx.5 analògica. També es pot fer circular l'aigua des del tanc BAIX fins al vas d'expansió activant la motobomba i obrint les electrovàlvules Vx.1 i Vx.2.

Totes les electrovàlvules tenen una vàlvula manual en paral·lel. Aquesta maqueta disposa d'un teclat d'operació amb pulsadors i selectors, aquest teclat ens ha de permetre posar la maqueta en funcionament.

#### 2.1.1.1 Sensors

Com elements sensorials hi ha:

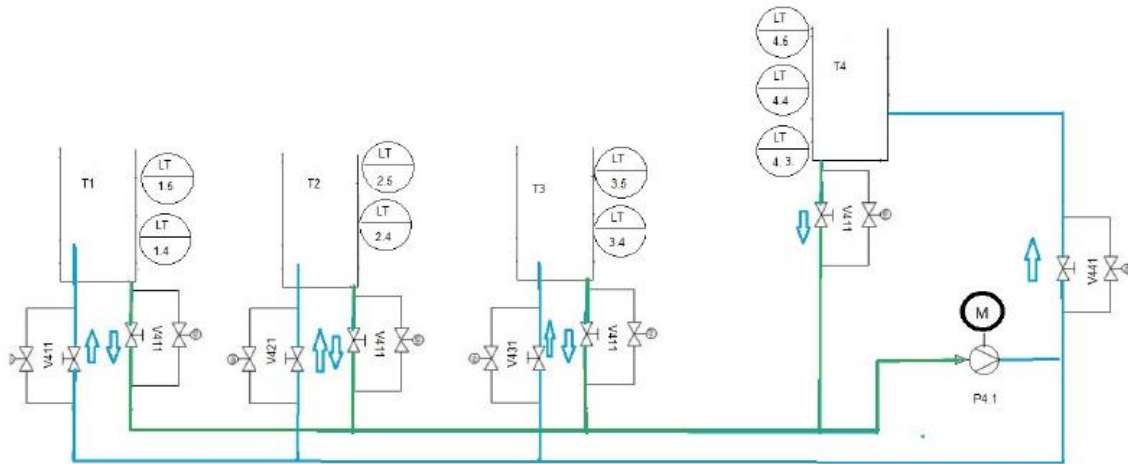
- 5 sensors lògics de nivell capacitatiu, dos estan situats al tanc superior, ALT, s'anomenen LTx.1 i LTx.2, i els altres tres estan situats al tanc inferior, BAIX, i s'anomenen LTx.3, LTx.4, LTx.5;
- Un sensor de nivell analògic, LTx.6, situat al tanc ALT o al BAIX, que ens permet conèixer el nivell d'aigua en el dipòsit;
- Un sensor de pressió, PTx.1, situat al vas de pressurització, que ens permet conèixer la pressió del recipient;
- Un sensor de flux, FTx.1, que ens permet conèixer el flux d'aigua que circula pel tub que comunica el dipòsit BAIX amb el dipòsit ALT.

#### 2.1.1.2 Actuadors

En quant a actuadors, es disposa de:

- Una motobomba, Px.1;
- 4 electrovàlvules normalment tancades distribuïdes al llarg de l'estació, Vx.1, Vx.2, Vx.3 i Vx.4.
- Una electrovàlvula motoritzada, Vx.5An, que permet regular el pas d'aigua entre els dipòsits ALT i BAIX.
- 5 vàlvules manuals en paral·lel amb les electrovàlvules.

### 2.1.2 Maqueta distribució



Imatge 7: Esquema maqueta distribució

#### 2.1.2.1 Sensors

Com elements sensorials hi ha:

- 8 sensors lògics de nivell capacitius, dos estan situats als tancs de les maquetes 1, 2, 3 i 4, i s'anomenen LTx.4 i LTx.5;

#### 2.1.2.2 Actuadors

En quant a actuadors, es disposa de:

- Una motobomba, P4.1
- 8 electrovàlvules normalment tancades distribuïdes al llarg de l'estació, V4i1 i V4i2, amb  $i = 1, 2, 3$  i 4. Les electrovàlvules 2 són de sortida d'aigua dels tancs de les estacions i les 1 són entrades.



### 2.1.3 Teclats d'operacions

#### 2.1.3.1 Teclat d'operacions de les maquetes bàsiques

La Imatge 8 mostra el teclat d'operació de la maqueta 1, els teclats de les altres maquetes bàsiques són iguals, només canvia la numeració. Per tenir accés a les entrades i sortides físiques, cal que l'interruptor OFF/ON estigui a ON.



Imatge 8: Teclat operacions maqueta 1

Com podem veure a la Imatge 8, el teclat d'operació disposa de:

- Un selector ON/OFF per alimentar dels mòduls d'entrades i sortides del PLC, els sensors i els actuadors.
- Un pulsador START que el podem fer servir per programa com a botó de marxa.
- Un selector A/M (Automàtic/Manual) .
- Un pulsador STOP que el podem fer servir per programa com a botó de parada.
- Un interruptor de parada d'emergència .
- Un LED ALARMA que podem programar perquè s'encengui quan hi hagi una alarma.
- Un LED POWER que per programa farem que indiqui quan el sistema està en marxa.

### 2.1.3.2 Teclat d'operacions de la maqueta distribució



Imatge 9: Teclat operacions maqueta distribució

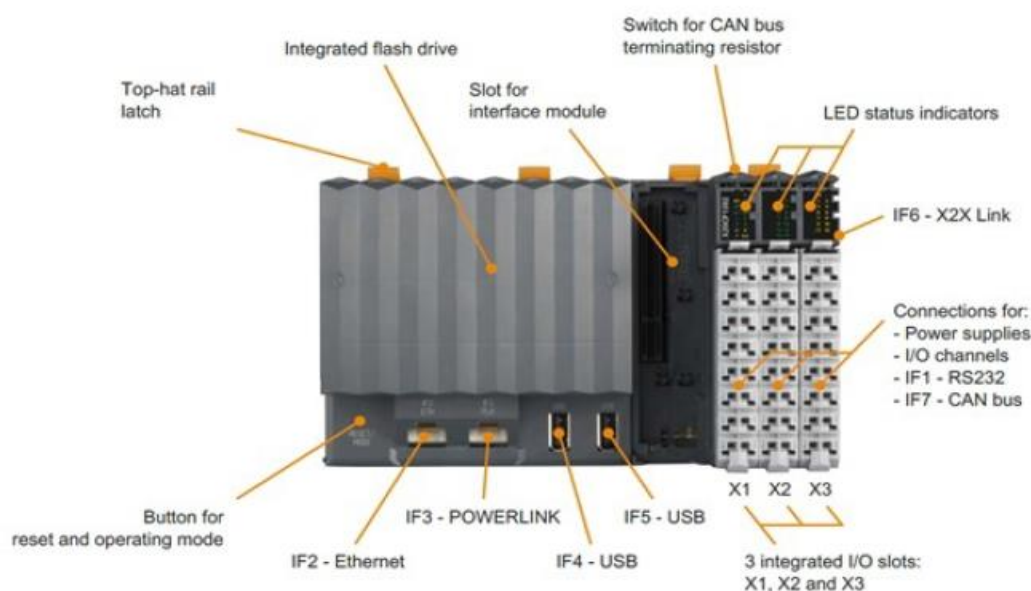


## 2.2 Descripció dels PLC's i els seus components

En aquest apartat es realitza una descripció de las característiques tant del autòmat com dels seus components y mòduls. Tots ells son del fabricant B&R INDUSTRIAL AUTOMATION [4].

### 2.2.1 PLC Maquetes bàsiques

Per la realització d'aquest projecte s'ha utilitzat com a PLC de les maquetes bàsiques 1,2 i 3, un autòmat del fabricant B&R INDUSTRIAL AUTOMATION X20CP1381 el qual es pot veure a la següent imatge:



Imatge 10: PLC B&R X20CP1381

Aquest autòmat te les següents característiques:

X20CP1381	X20 CPU, with integrated I/O, x86-200, 128 MB DDR3 RAM, 16 kB FRAM, 1 GB onboard flash drive, 1 insert slot for X20 interface modules, 2 USB interfaces, 1 RS232 interface, 1 CAN bus interface, 1 POWERLINK interface, 1 Ethernet interface 10/100BASE-T, 14 digital inputs, 24 VDC, sink, 4 digital inputs, 2 $\mu$ s, 24 VDC, sink, 4 digital outputs, 24 VDC, 0.5 A, source, 4 digital outputs, 2 $\mu$ s, 24 VDC, 0.2 A, 4 digital inputs/outputs, 24 VDC, 0.5 A, 2 analog inputs $\pm 10$ V or 0 to 20 mA / 4 to 20 mA, 1 PT1000 instead of an analog input, includes power supply module, 3x terminal block X20TB1F, slot cover and X20 end cover plate X20AC0SR1 (right) included
-----------	---

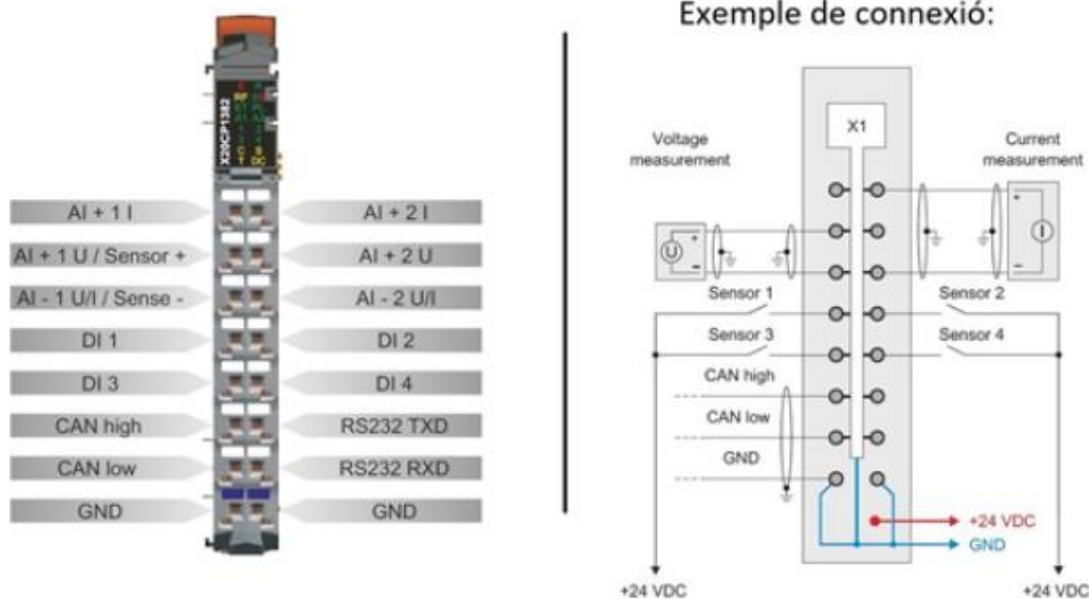
Imatge 11: Característiques PLC X20CP1381

El PLC B&R X20CP1381 porta integrats 3 mòduls d'entrades i sortides (X1, X2 i X3) més un mòdul d'expansió, que es descriuen a continuació:

### 2.2.1.1 Mòdul X1

El mòdul X1 (veure Imatge 12) disposa de:

- 2 entrades analògiques, les quals poden llegir senyals de corrent o tensió en funció de la connexió que es realitzi. Per exemple, en el cas de l'entrada analògica de tensió la connexió es faria entre els terminals AnalogInput+1U i AnalogInput-1U, i una entrada analògica de corrent es faria amb els terminals AnalogInput+2I i AnalogInput-2U/I.
- 4 entrades lògiques a 24 V DigitalInput1, DigitalInput2, DigitalInput3 i DigitalInput4.
- 2 bussos de comunicació CAN i RS232.

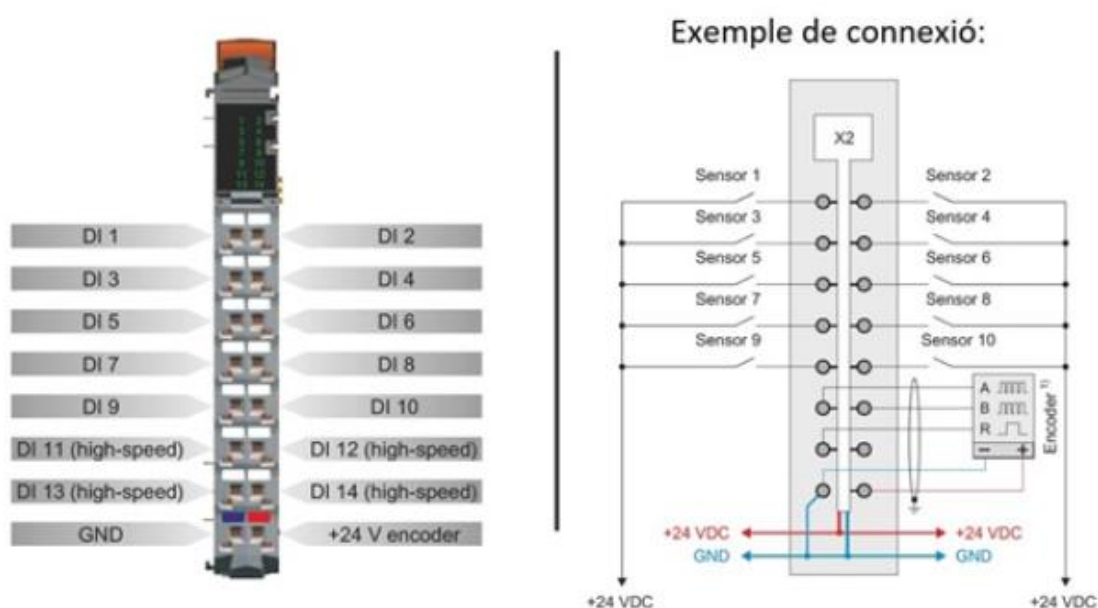


Imatge 12: Mòdul E/S X1

### 2.2.1.2 Mòdul X2

El mòdul X2 (veure Imatge 13) disposa de:

- 10 entrades lògiques a 24 normals de DigitalInput1 a DigitalInput10;
- 2 entrades lògiques d'alta velocitat DigitalInput11, DigitalInput12, DigitalInput13 i DigitalInput14, a les quals es poden connectar encoders de la manera que es mostra a la Imatge 13.

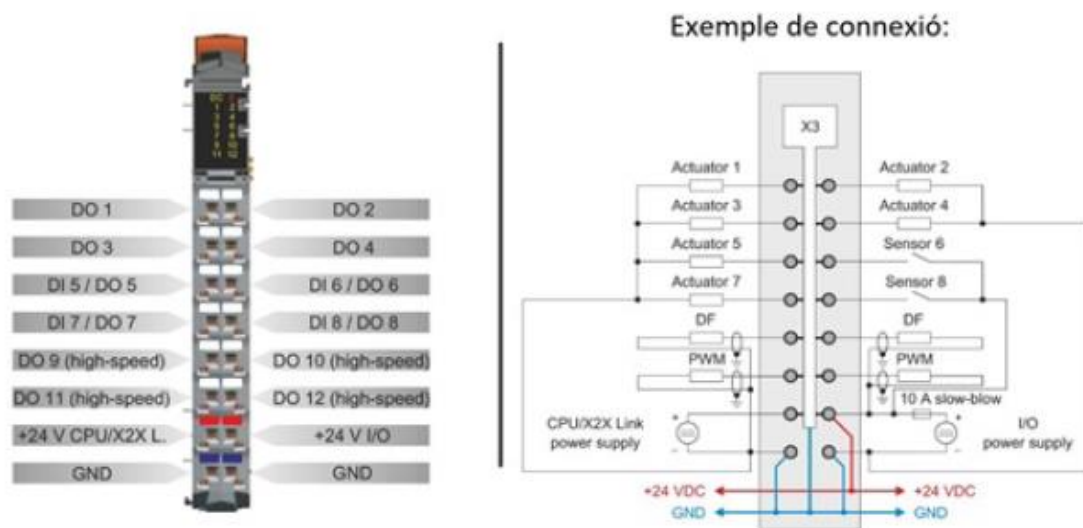


Imatge 13: Mòdul E/S X2

### 2.2.1.3 Mòdul X3

El mòdul X3 (veure Imatge 14) disposa de:

- 4 sortides lògiques: DigitalOutput1 a DigitalOutput4.
- 4 entrades/sortides configurables : DigitalInput5 / DigitalOutput5 a DigitalInput8 / DigitalOutput8.
- 4 sortides lògiques d'alta velocitat: DigitalOutput9 a DigitalOutput12.
- Connexions a una font d'alimentació, l'alimentació a la CPU i a les entrades/sortides pot ser independent.

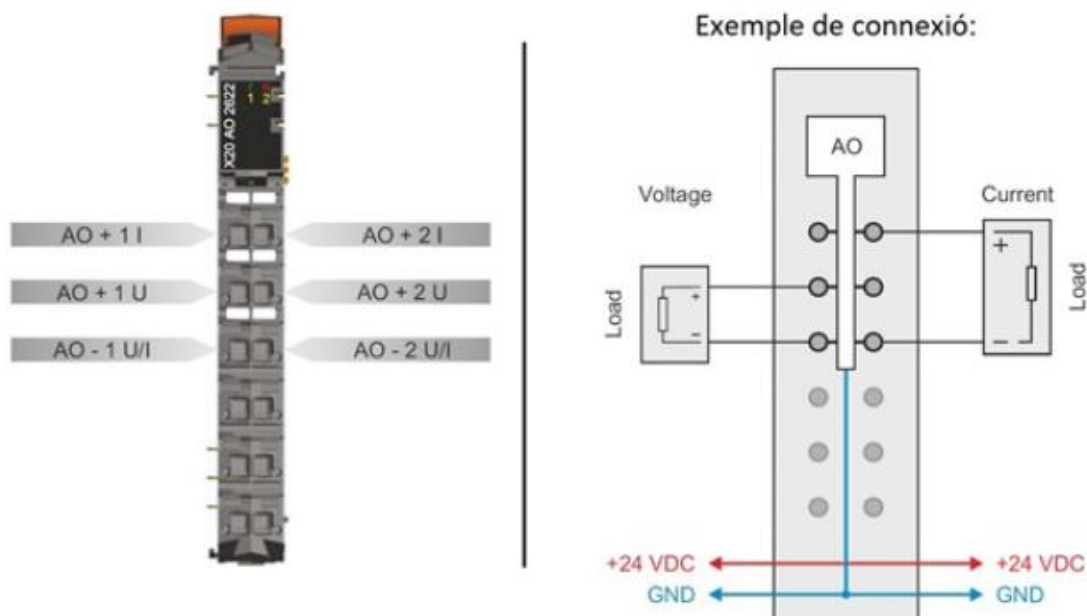


Imatge 14: Mòdul E/S X3

#### 2.2.1.4 Mòdul X4

El mòdul X4, és un mòdul incorporat de codi X20AO2622 (veure Imatge 15), disposa de:

- 2 sortides analògiques de corrent o tensió.



Imatge 15: Mòdul sortides analògiques X4

### 2.2.1.5 Assignació d'entrades i sortides

La següents taules mostren a quin mòdul i a quin canal estan connectats cada sensor, actuador i element del teclat d'operacions en els mòduls del PLC.

Taula 2: Assignació Entrades PLC X20CP1381

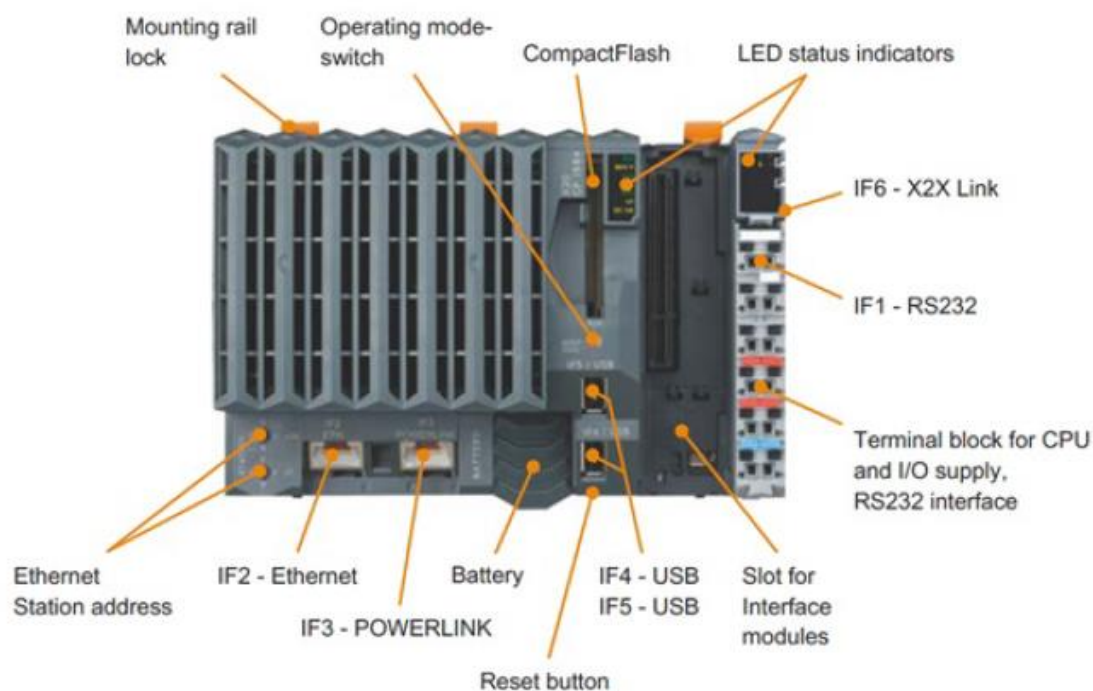
Mòdul	Canal	Símbol	Descripció
X1	AnalogInput01	PTx.1	Sensor de pressió
X1	AnalogInput02	LTx.6	Sensor de nivell analògic
X2	DigitalInput01	LTx.1	Sensor nivell baix tanc ALT
X2	DigitalInput02	LTx.2	Sensor nivell alt tanc ALT
X2	DigitalInput03	LTx.3	Sensor nivell baix tanc BAIX
X2	DigitalInput04	LTx.4	Sensor nivell mig tanc BAIX
X2	DigitalInput05	LTx.5	Sensor nivell alt tanc BAIX
X2	DigitalInput06	OFF/ON	Selector tensió maqueta
X2	DigitalInput07	START	Polsador de marxa
X2	DigitalInput08	A/M	Selector Auto/Manual
X2	DigitalInput09	STOP	Polsador de parada
X2	DigitalInput10	RESET	Polsador de reset
X2	DigitalInput11	FTx.1	Sensor de cabal

Taula 3: Assignació Sortides PLC X20CP1381

Mòdul	Canal	Símbol	Descripció
X3	DigitalOutput01	Vx.1	Electrovàlvula 1
X3	DigitalOutput02	Vx.2	Electrovàlvula 2
X3	DigitalOutput03	Vx.3	Electrovàlvula 3
X3	DigitalOutput04	Vx.4	Electrovàlvula 4
X3	DigitalOutput05	POWER	LED de marxa
X3	DigitalOutput06	ALARM	LED d'alarma
X4	AnalogOutput01	Vx.5 An	Electrovàlvula proporcional
X4	AnalogOutput02	Px.1	Motobomba

### 2.2.2 PLC Maqueta distribució

A més a més dels PLC's anteriors, la maqueta de distribució té un PLC diferent que també es del fabricant B&R INDUSTRIAL AUTOMATION però el model es el X20CP1583 el qual es pot veure a la següent imatge:



Imatge 16: PLC B&R X20CP1583

Aquest autòmat te les següents característiques:

X20CP1583	X20 CPU, Atom 333 MHz Intel compatible, 128 MB DDR2 RAM, 1 MB SRAM, removable application memory: CompactFlash, 1 insert slot for X20 interface modules, 2 USB interfaces, 1 RS232 interface, 1 Ethernet interface 10/100/1000BASE-T, 1 POWERLINK interface, including power supply module, 1x terminal block X20TB12, slot cover and X20 end cover plate X20AC0SR1 (right) included, order application memory separately!
-----------	--

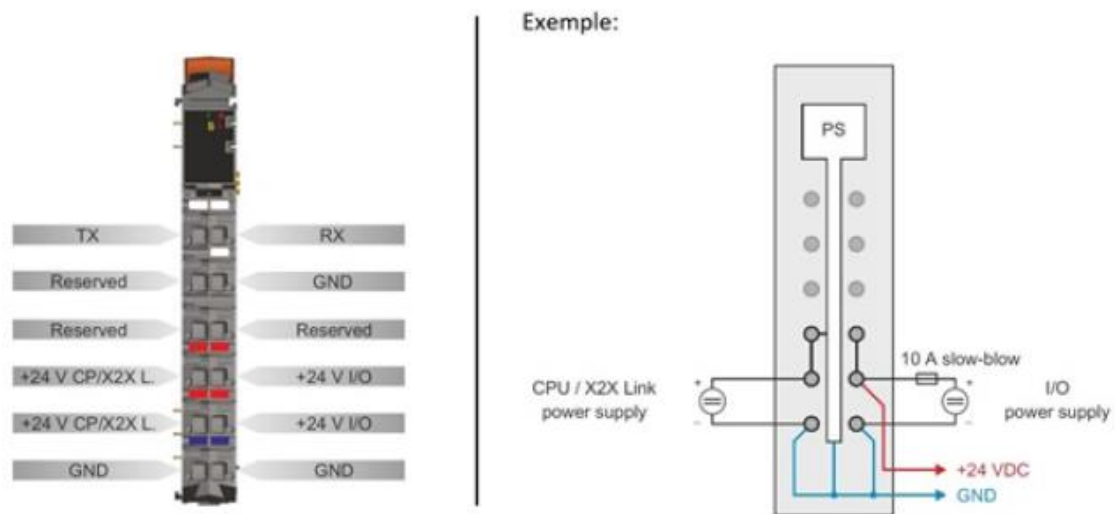
Imatge 17: Característiques PLC X20CP1583

El PLC X20CP1583 disposa d'un mòdul bàsic que inclou l'alimentació externa, i les comunicacions X2X Link i RS232 (veure Imatge 18).

S'han incorporat dos mòduls, el X20DIF371 (veure Imatge 19) i el X20DOF322 (veure Imatge 20), que disposen de 16 entrades digitals i 16 sortides digitals respectivament. A continuació es mostra com son els mòduls i com es connecten.

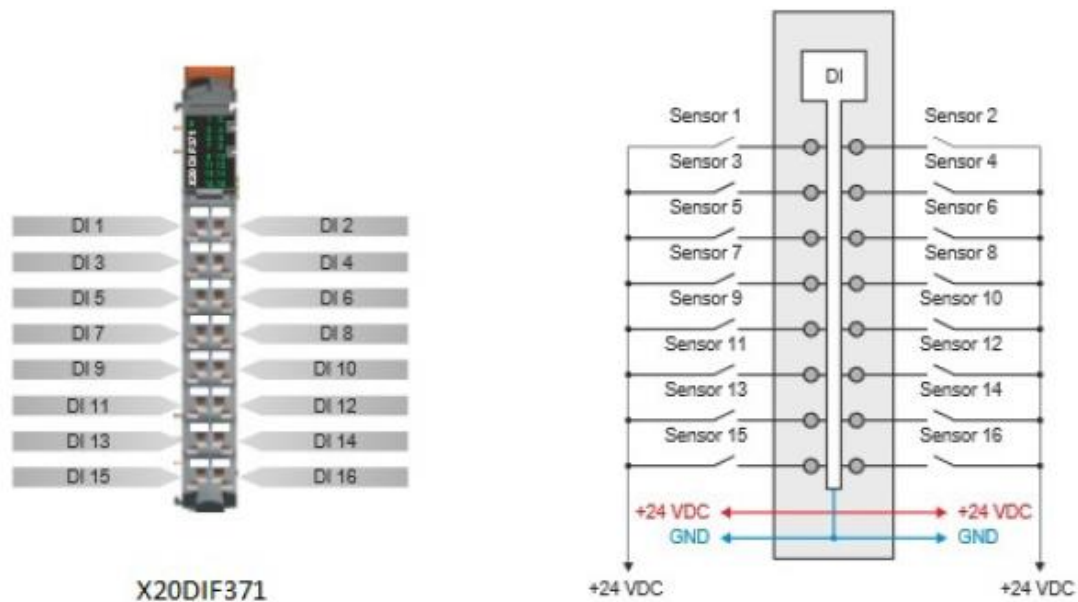


### 2.2.2.1 Mòdul d'alimentació i comunicacions



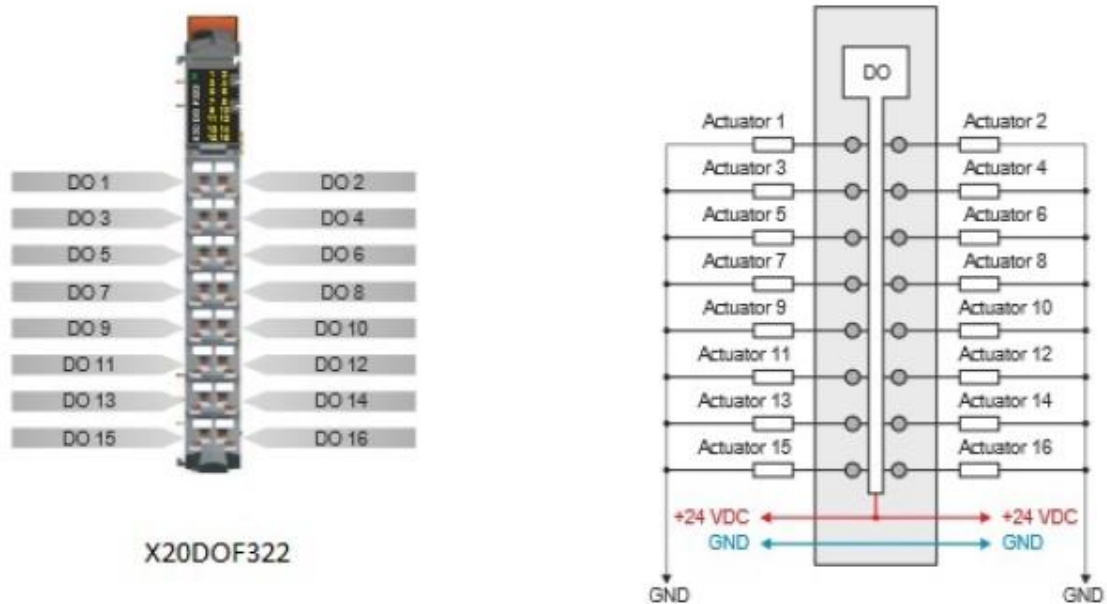
Imatge 18: Mòdul alimentació i comunicacions

### 2.2.2.2 Mòdul X20DIF371



Imatge 19: Mòdul X20DIF371

### 2.2.2.3 Mòdul X2DOF322



Imatge 20: Mòdul X2DOF322



### 2.2.2.4 Assignació d'entrades i sortides

La següents taules mostren a quin mòdul i a quin canal estan connectats cada sensor, actuator i element del teclat d'operacions en els mòduls del PLC.

Taula 4: Assignació Entrades PLC X20CP1583

Mòdul	Canal	Símbol	Descripció
X20DIF371	DigitalInput01	LT15	Nivell alt Tanc 1
X20DIF371	DigitalInput02	LT14	Nivell baix Tanc 1
X20DIF371	DigitalInput03	LT25	Nivell alt Tanc 2
X20DIF371	DigitalInput04	LT24	Nivell baix Tanc 2
X20DIF371	DigitalInput05	LT34	Nivell baix Tanc 3
X20DIF371	DigitalInput06	LT35	Nivell alt Tanc 3
X20DIF371	DigitalInput07	LT44	Nivell mig Tanc 4
X20DIF371	DigitalInput08	LT45	Nivell alt Tanc 4
X20DIF371	DigitalInput09	START	Polsador de marxa
X20DIF371	DigitalInput10	STOP	Polsador de parada
X20DIF371	DigitalInput11	AUTO_MAN	Selector Auto/Manual
X20DIF371	DigitalInput12	RESET	Polsador de Reset
X20DIF371	DigitalInput13	I4	Reserva
X20DIF371	DigitalInput14	I5	Reserva
X20DIF371	DigitalInput15	I6	Reserva
X20DIF371	DigitalInput16	LT43	Nivell baix Tanc 4

Taula 5: Assignació Sortides PLC X20CP1583

Mòdul	Canal	Símbol	Descripció
X20DOF322	DigitalOutput01	V421	Vàlvula entrada Tanc 2
X20DOF322	DigitalOutput02	V422	Vàlvula sortida Tanc 2
X20DOF322	DigitalOutput03	V431	Vàlvula entrada Tanc 3
X20DOF322	DigitalOutput04	V432	Vàlvula sortida Tanc 3
X20DOF322	DigitalOutput05	V411	Vàlvula entrada Tanc 1
X20DOF322	DigitalOutput06	V412	Vàlvula sortida Tanc 1
X20DOF322	DigitalOutput07	V441	Vàlvula entrada Tanc 4
X20DOF322	DigitalOutput08	V442	Vàlvula sortida Tanc 4
X20DOF322	DigitalOutput09	L_START	LED de marxa
X20DOF322	DigitalOutput10	L_RESET	LED d'alarma
X20DOF322	DigitalOutput11	Q1	Reserva
X20DOF322	DigitalOutput12	Q2	Reserva
X20DOF322	DigitalOutput13	Q3	Reserva
X20DOF322	DigitalOutput14	Q4	Reserva
X20DOF322	DigitalOutput15	LT43	Nivell baix Tanc 4
X20DOF322	DigitalOutput16	PUMP	Bomba

## 2.3 Pantalla tàctil

La pantalla tàctil utilitzada en aquest projecte es també del mateix fabricant que els autòmats programables, B&R AUTOMATION, i es el model 6PPT30.0702-20W [4], a la següent imatge es pot veure com es:



Imatge 21: Pantalla HMI

Les característiques principals de la pantalla son les següents:

Model number	Short description
	<b>Power Panel T30</b>
6PPT30.0702-20W	Power Panel T30, 7.0", landscape format. CPU and memory: 600 MHz (ARM Cortex-A8), 512 MB onboard flash drive. Display and touch screen: 7.0", 800 x 480 (WVGA) resolution, analog resistive touch screen, landscape format, aluminum white pin-stripe. Interfaces: 2x Ethernet 10/100 Mbit/s (integrated switch), 2x USB 2.0. Client software: Integrated service page, VNC client, embedded web browser.

Imatge 22: Característiques de la pantalla HMI

### 3. ENTORN DE PROGRAMACIÓ

A continuació es mostra el programa utilitzat per la realització de tot el projecte, també s'explicarà com començar una senzilla programació per a cada part que integra el projecte.

Amb aquest programa es pot tant programar les maquetes, como realitzar la visualització del procés i el control d'alguns de los paràmetres, i preparar el software per configurar tant els clients com el servidor OPC per llegir i escriure les variables i senyals de cada un dels autòmats per intercanviar informació entre ells.

#### 3.1 B&R Automation Studio

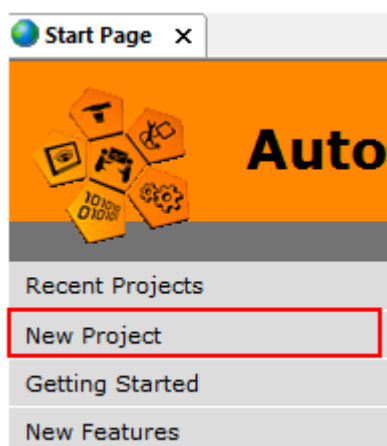
Aquest software es l'utilitzat per programar els PLC's, per dissenyar i programar la pantalla de visualització i per configurar el servidor i clients OPC, és un tot en un, i això es un punt a favor ja que facilita la feina al no haver de tenir diferents programes per fer les dues coses.



Imatge 23: Versió utilitzada Automation Studio

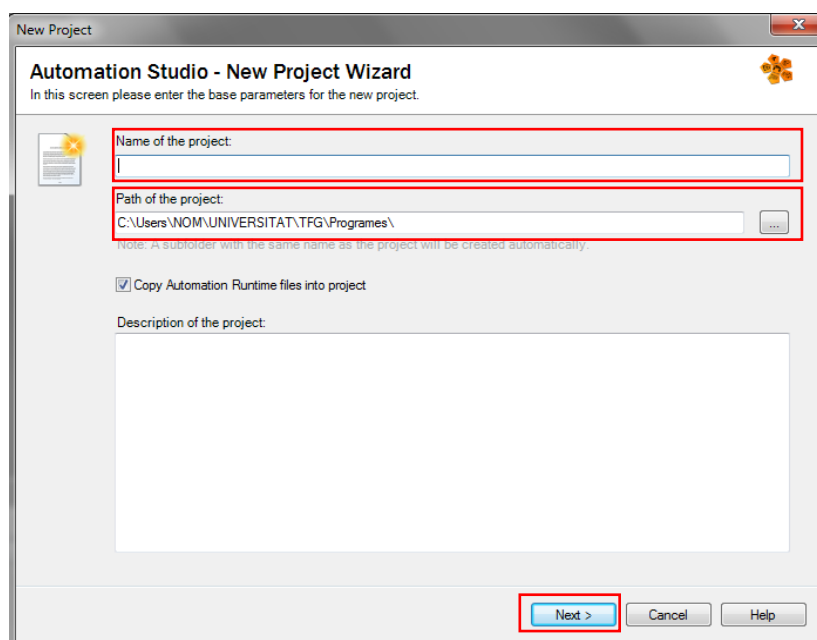
##### 3.1.1 Creació d'un nou programa

Un cop instal·lat el programa en el nostre ordinador, l'iniciem i fem clic a 'New Project'.



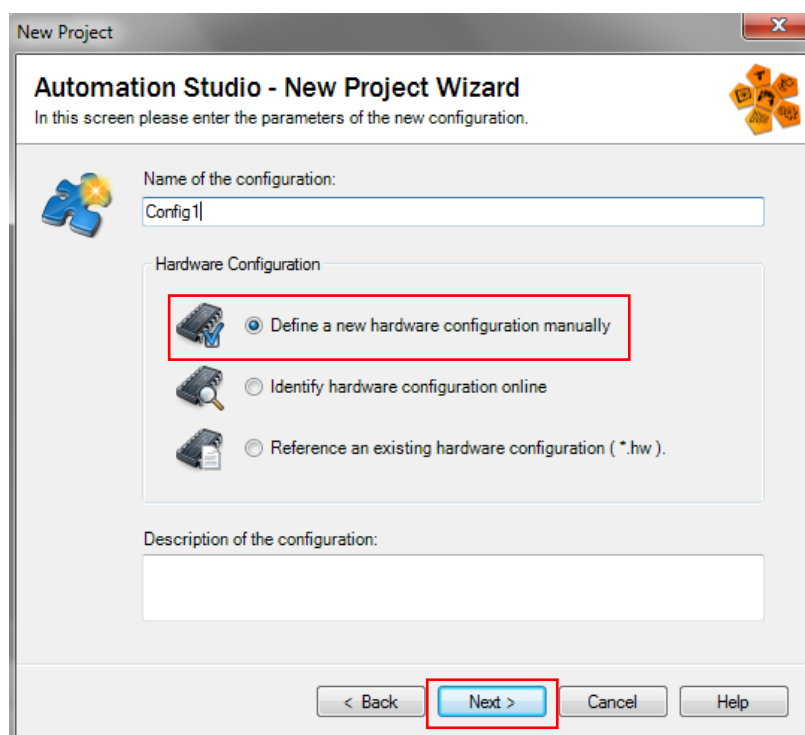
Imatge 24: Nou Projecte AS

Cliquem a 'An empty projecte' i se'ns obrirà una nova finestra on podrem escollir el nom que vulguem per el nostre projecte i la ubicació on aquest es guardarà (veure Imatge 25) i fem clic a Next.



Imatge 25: Nom i ubicació nou projecte AS

A continuació se'ns obrirà una nova finestra on haurem d'escollir el nom que li volem donar a la nova configuració i la manera que volem configurar el nou hardware del nostre projecte.



Imatge 26: Escollir configuració hardware AS

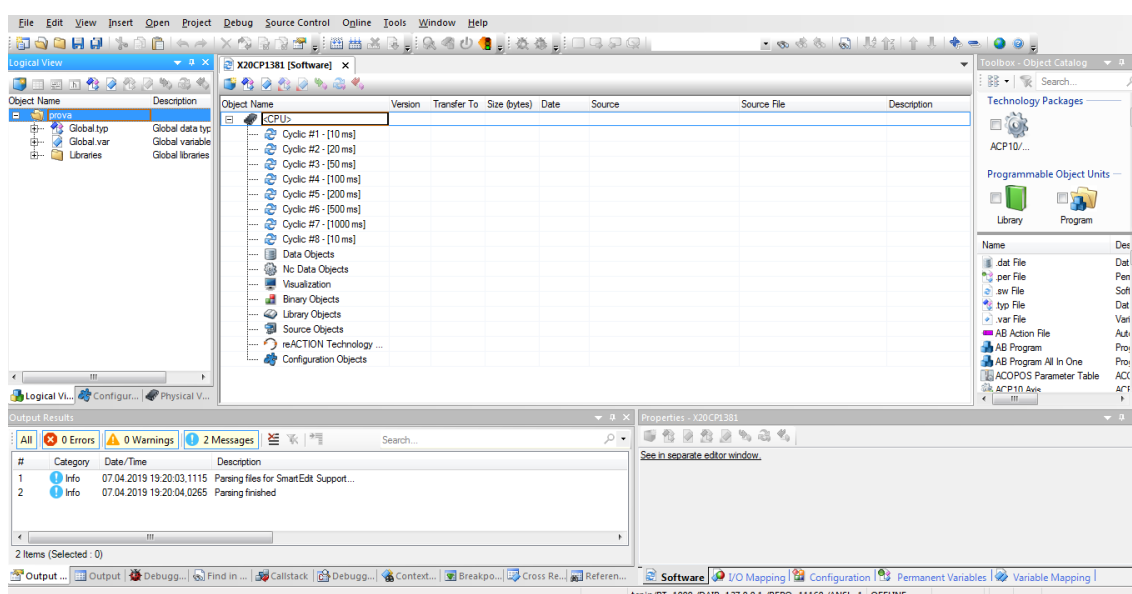
La forma de configurar el nou hardware es pot fer de diferents maneres, les dues principals son:

- Triar manualment el hardware dels que disposem a partir d'una llista d'elements.
- Connectar-nos directament al PLC i crear la configuració automàticament.

En el nostre cas em escollit la primera opció, ja que al crear el nou projecte no teníem possibilitat de connectar-nos amb el PLC in situ. Per tant fem clic a la opció *'Define a new hardware configuration manually'* i a continuació clic a Next.

Per acabar la creació del nou projecte, busquem a la llista de components hardware el nostre PLC, depenent de quina maqueta sigui buscarem el PLC X20CP1381 o bé el X20CP1583, i finalitzem clicant a *Finish*.

Un cop creat el projecte se'ns obrirà la pagina del entorn de treball (veure Imatge 27).



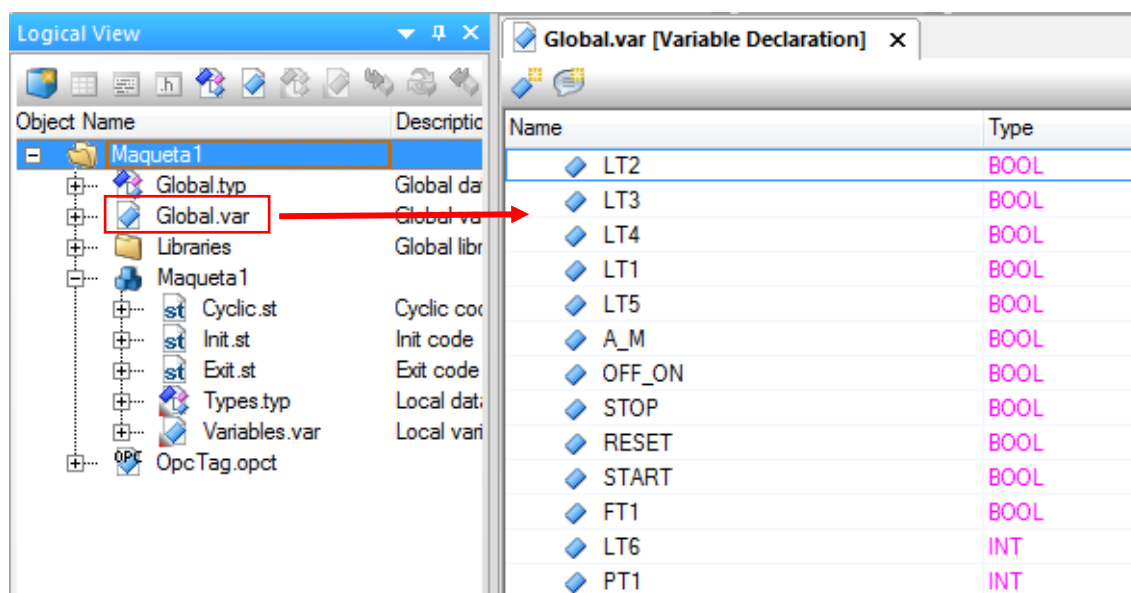
Imatge 27: Entorn de treball del AS

A la part esquerra hi ha tres pestanyes, la primera es la *Logic View*, aquí es on podem definir les nostres variables globals, afegir els programes, insertar el disseny de la pantalla tàctil (mes endavant veurem com fer-ho) i tot el que estigui relacionat amb la programació lògica.

La segona pestanya es la *Configuration View*, com diu el seu pròpi nom es on podrem configurar i modificar tot el que estigui relacionat amb el nostre programa.

Per ultim la tercera pestanya es la *Physical View*, en aquesta es on es troba tot l'ho relacionat amb el nostre hardware i on podrem configurar les entrades i sortides, el port Ethernet i modificar el software i hardware del pròpi PLC.

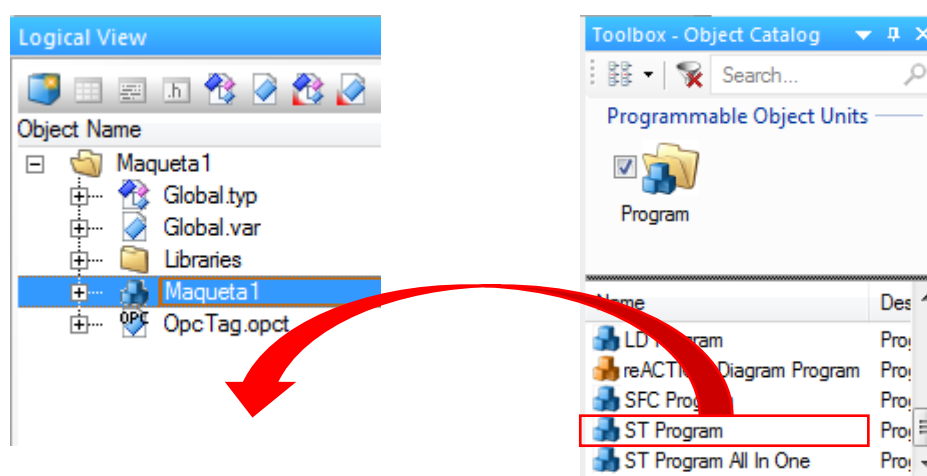
Per començar a treballar sobre el nostre programari, el primer que fem es declarar totes les variables globals del procés a controlar, per fer-ho anem a la pestanya de *Logic View*, i fem doble clic sobre *Global*, se'ns obrirà una finestra on podrem afegir totes les variables amb el seu nom i el tipus que es (BOOL, INT, DINT, REAL, etc...) tal i com es mostra a la Imatge 28.



Imatge 28: Declaració variables globals

Seguidament passem a afegir un mòdul de programació on estarà el programa a executar per el PLC i les variables locals. Abans però em de definir quin llenguatge de programació utilitzarem i per això com hem dit anteriorment tenim diferents llenguatges, l'escollit per aquest projecte ha estat principalment el de text estructurat (ST), tot i que s'han utilitzat també d'altres.

Per afegir el mòdul de programa em de buscar-lo a la finestra de la dreta del entorn de programació 'Toolbox - Object Catalog' i arrastrar-lo fins a la finestra *Logic* (veure Imatge 29).



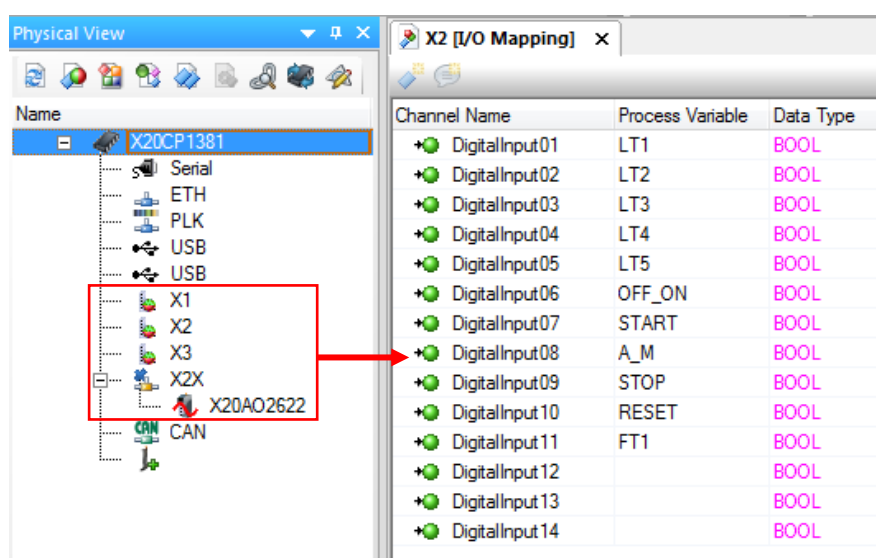
Imatge 29: Afegir mòdul de programa

Ja afegit el nostre programa si obrim el desplegable, veiem que hi ha diferents arxius:

- *Cyclic*: Es on escriurem el nostre programa i que s'executarà cíclicament.
- *Init*: Es el programa que s'executarà només el primer cop que arrenqui el PLC.
- *Exit*: Es el programa que s'executarà l'últim cicle abans d'acabar.

I després hi ha les variables locals, les variables que afegim aquí només seran visibles per aquest programa en concret i cap altre programa podrà llegir-les o escriure-les.

Per poder configurar les entrades i sortides físiques del nostre PLC hem de dirigir-nos a la pestanya *Physical*, com em dit anteriorment, i obrir qualsevol dels mòduls X1, X2, X3 o X20 i allà podrem afegir cada variable global a l'entrada o sortida corresponent.



Imatge 30: Assignació variables a E/S

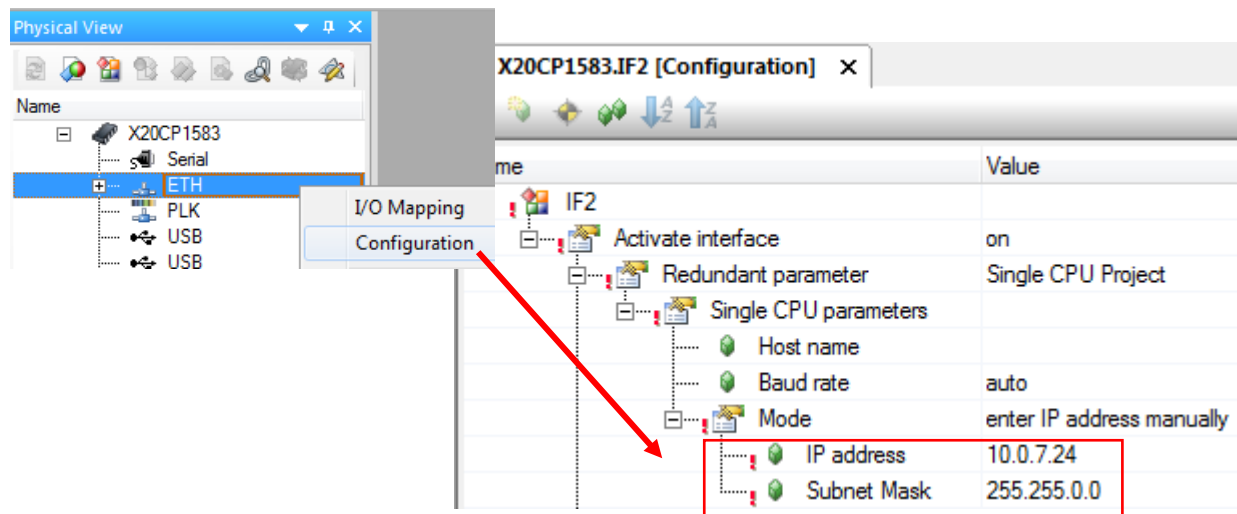
Per assignar a cada entrada o sortida, una variable global o local prèviament creada, fem doble clic sobre el requadre corresponent de la columna *Process Variable* i se'ns obrirà una finestra on es permetrà buscar la variable que desitgem.

Quan fem el mapejat de les E/S hem de tenir en compte que a cada entrada i sortida digital el tipus de variable haurà de ser del tipus **BOOL** i per les entrades i sortides analògiques haurà de ser del tipus **INT**.

Un altre punt important és la configuració del mòdul físic Ethernet per poder comunicar-nos amb el PLC.

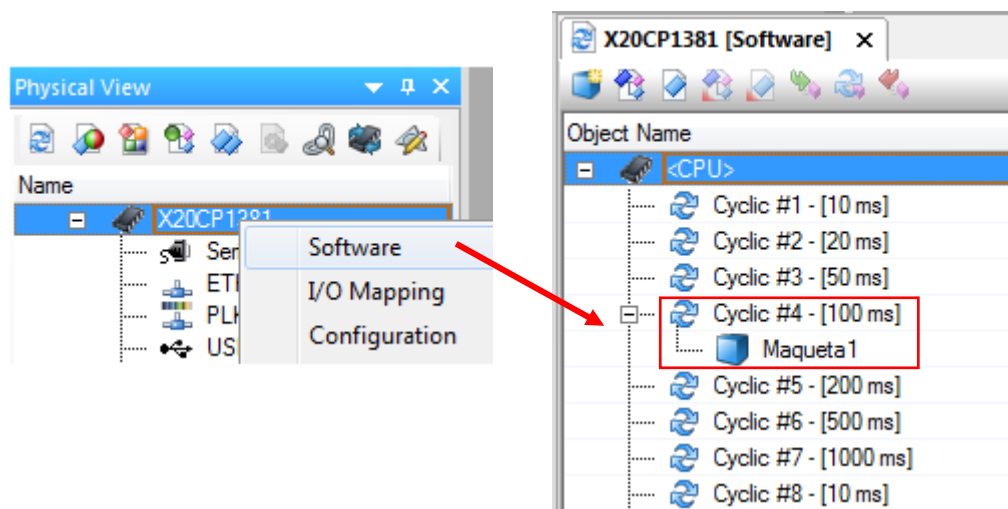
Des de la pestanya *Physical* i fent clic amb el botó dret sobre el mòdul ETH, seleccionem *Configuration* i a la pestanya de configuració del mòdul que se'ns obre podem configurar la IP i la mascara de subred del autòmat tal i com es mostra a la Imatge 31.





Imatge 31: Configuració mòdul Ethernet

Un cop creat el programa, les variables tant globals com locals i configurat el hardware només ens queda configurar la part de software de la CPU i assignar els diferents mòduls de programa a cada cicle de treball que volem que s'executi, per fer-ho anem a la pestanya *Physical* i fent clic amb el boto dret sobre el PLC, seleccionem *Software* i a la finestra que se'n obra podem afegir cada programa al cicle de temps que vulguem, com es mostra a la Imatge 32.



Imatge 32: Assignació programes al cicles de treball



### 3.1.2 Configuració d'un sistema de visualització

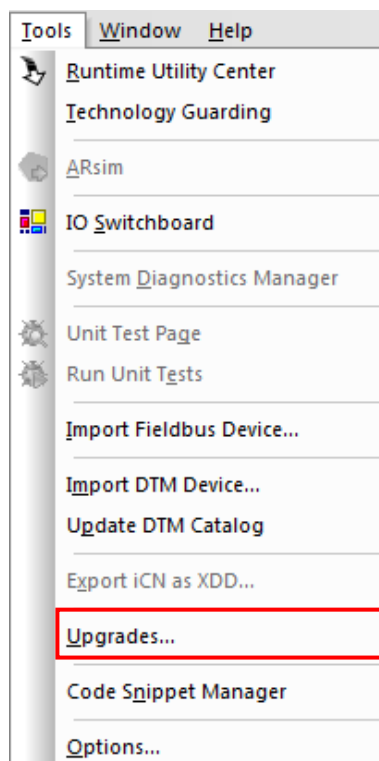
A continuació es mostra els passos a seguir per configurar el software i el hardware per poder programar i dissenyar la pantalla.

#### 3.1.2.1 Actualització de la imatge i el hardware de la pantalla

Abans de començar a configurar res, hem d'assegurar-nos que la versió de la imatge i del hardware de la pantalla es la corresponent per poder treballar-hi.

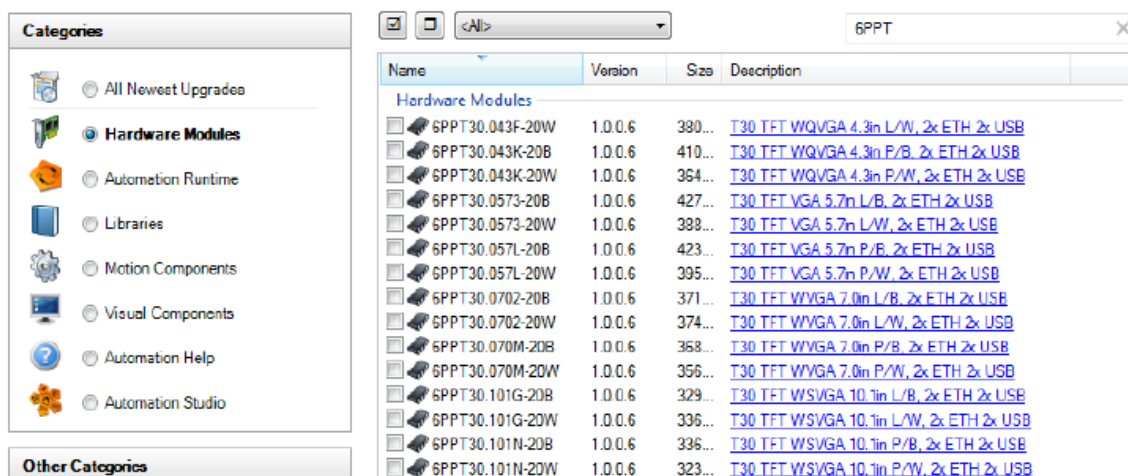
Les T30 / T50 "arriben de fàbrica" amb una imatge per defecte la versió no necessàriament coincideixi amb la versió del projecte que s'està desenvolupant a AS. Fent ús d'AS i un USB és possible actualitzar aquesta imatge a la versió del projecte en desenvolupament i a la vegada carregar un logo d'arrencada i la configuració de la T30 / T50 que determinarà la seva forma de treball.

Si no es té el maquinari de la T30 / T50 instal·lat en AS, o es vol treballar amb l'última versió disponible, el primer pas a realitzar és la descàrrega del fitxer de maquinari de la T30 / T50 del servidor de B & R. Per a això cal anar al menú principal d'AS i seleccionar Tools → Upgrades.



Imatge 33: Actualització del programari de  
AS

Al menú de Upgrades primer s'ha de descarregar l'actualització de maquinari corresponent a la T30 / T50 que s'utilitzarà (o s'usa) en el projecte (present en la categoria Maquinari Modules).

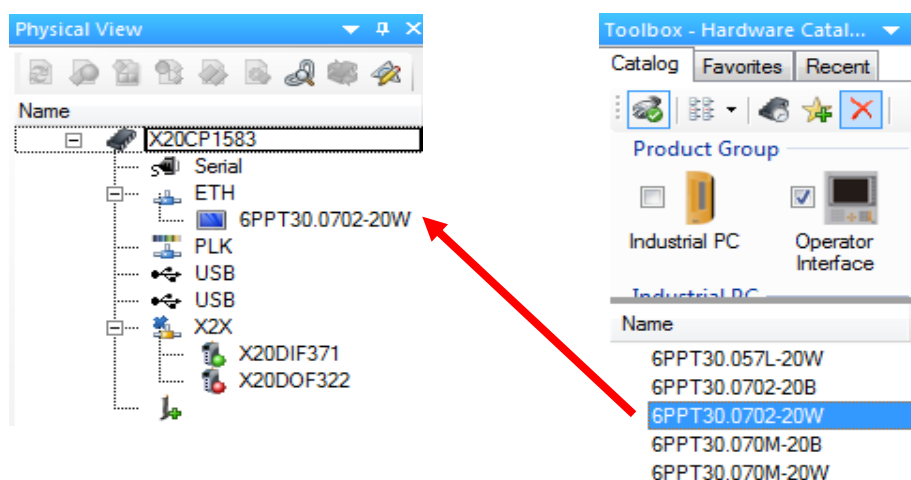


Imatge 34: Pantalla d'actualització d'AS

Aquesta actualització de maquinari funciona de la mateixa que una altra actualització de qualsevol altre mòdul de B&R, com pot ser la perifèria de x2x, i el seu contingut és la descripció de maquinari per AS entre d'altres. Per defecte en instal·lar AS s'instal·la al seu torn una primera imatge de les T30 (comuna per a tota la família de la T30). Aquesta s'instal·la a la ruta "C:\BrAutomation\AS\PPT\30".

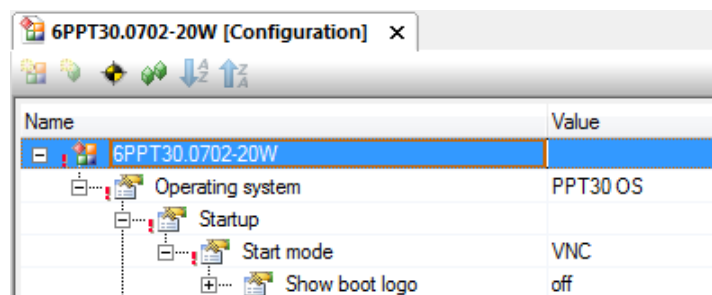
### 3.1.2.2 Afegir i configurar el terminal al projecte de Automation Studio

Primer de tot com ja hem fet anteriorment hem de buscar la pantalla a dintre la finestra del catàleg d'objectes ubicada a la dreta de tot de l'entorn de treball i arrossegar-la fins al nostre mòdul ETH ja que aquesta estarà connectada a través d'un cable Ethernet, els passos descrits es mostren a la Imatge 35.



Imatge 35: Afegir hardware pantalla tàtil

Un cop tenim el hardware actualitzat amb la pantalla que penja del mòdul ETH ja podem configurar-li els paràmetres necessaris per poder treballar-hi. Per fer-ho, fem clic amb el botó dret sobre la pantalla i anem a “Configuració”, amb això se’ns obre la pantalla de configuració de la pantalla on podem modificar els camps necessaris.



Imatge 36: Configuració Startup pantalla

El primer paràmetre que hem de modificar es el de l'arrencada de la pantalla, de quina manera serà, podem escollir entre tres opcions, Servicepage, VNC o Web, en el nostre cas seleccionarem la opció VNC (veure Imatge 36).

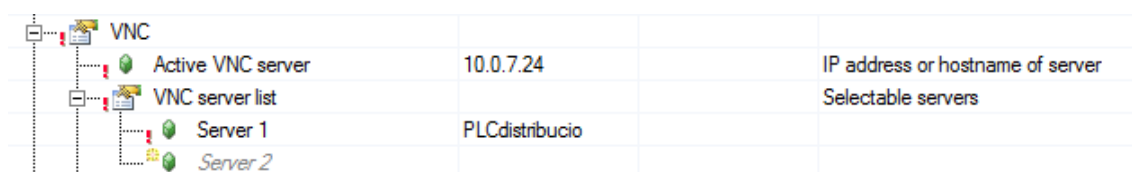
La part de la connexió la deixem tal qual i que ens detecti i assigni el programa la IP que té la pantalla.

Fet això cal assignar-li a la T30 / T50 una imatge de totes les disponibles. De nou a la vista de configuració s'ha d'anar a l'apartat *Device Files* i triar aquesta imatge a l'entrada PPT image.



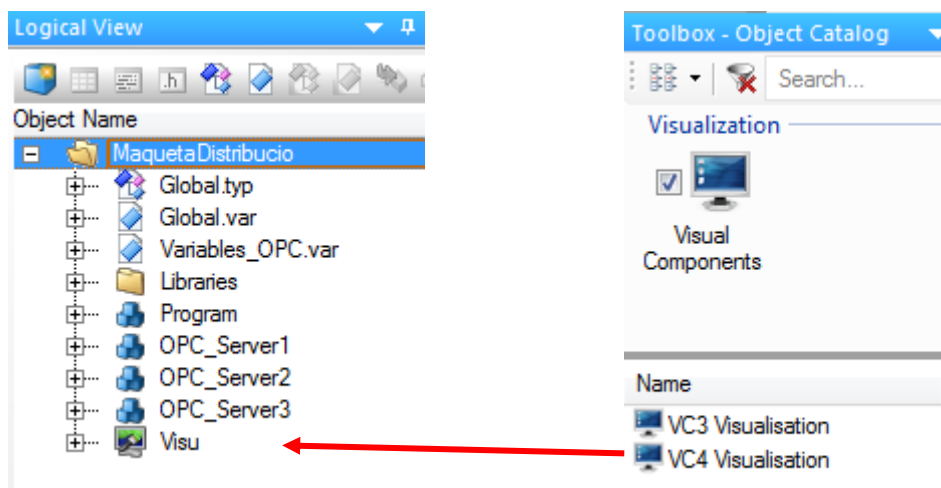
Imatge 37: Assignar arxiu d'imatge a la pantalla

A continuació a la pestanya VNC tenim dues opcions que hem de configurar, la primera es la de la direcció IP on està el servidor VNC que en el nostre projecte està ubicat a l'autòmat de la maqueta de distribució i per tant posem la IP que té assignada el PLC, i com a segon hem d'assignar-li un nom al servidor, en aquest cas s'ha escollit PLCdistribucio.



Imatge 38: Configuració IP VNC Server pantalla

Finalment només ens queda afegir a la finestra *Logic* l'element de visualització per poder dissenyar-lo i programar-lo. Per això busquem a la finestra d'objectes l'element que desitgem, en el nostre cas *VC4 Visualisation* i l'arrastrem (veure Imatge 39).

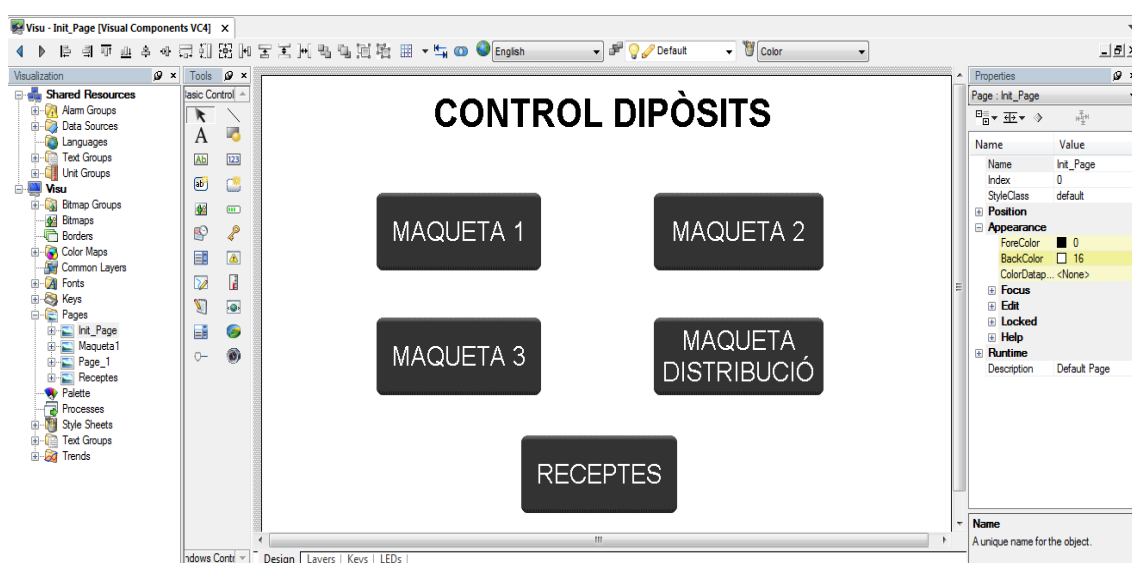


Imatge 39: Afegir element de visualització

### 3.1.2.3 Disseny de la pantalla

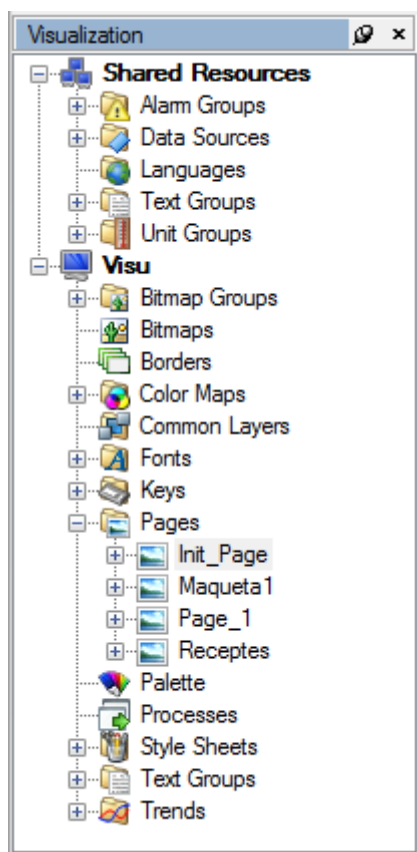
A continuació s'explica com començar a dissenyar una pantalla de visualització, descrivint els elements i passos inicials a tenir en compte.

Per començar, fem doble clic sobre l'element de visualització prèviament afegit, en el nostre cas "Visu", i se'ns obrirà l'espai de treball per dissenyar les pantalles i afegir els elements necessaris.

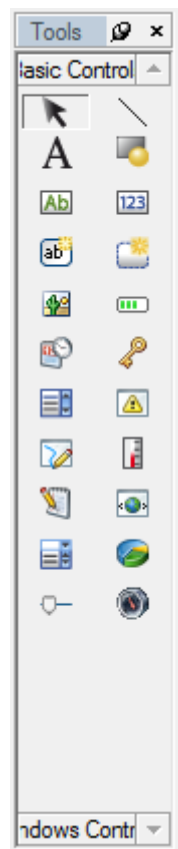


Imatge 40: Espai de treball, disseny de visualitzacions

A la part esquerra de la pantalla, ens apareixen dues finestres, la primera es on es mostren tots els elements que conté el nostre disseny com ara les imatges, les pàgines, els estils de text i colors, etc... (veure Imatge 42) on cada un d'ells és configurable i únic per cada disseny i la següent finestra es la de les eines on hi ha tots els objectes que podem utilitzar per crear les diferents pantalles com ara els botons, requadres de text, labels, figures, etc... (veure Imatge 41).



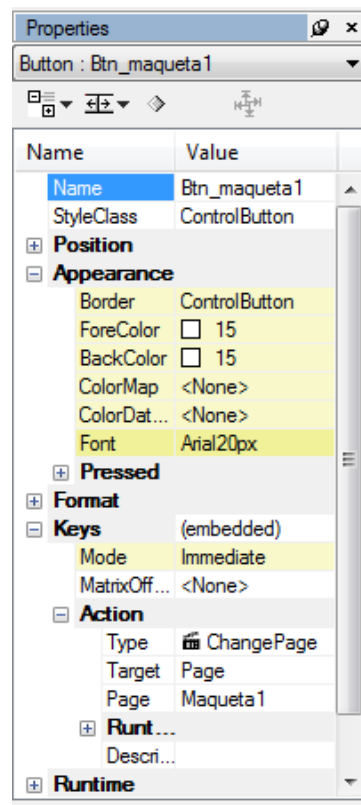
Imatge 42: Elements visualització



Imatge 41: Objectes visualització

I a la part dreta de la pantalla es on trobem la finestra de les propietats de cada objecte o element que tinguem al nostre disseny (veure Imatge 43). Els més destacables són el nom, el color de fons i del text, l'estil de lletra, el format i la variable de control de l'objecte.

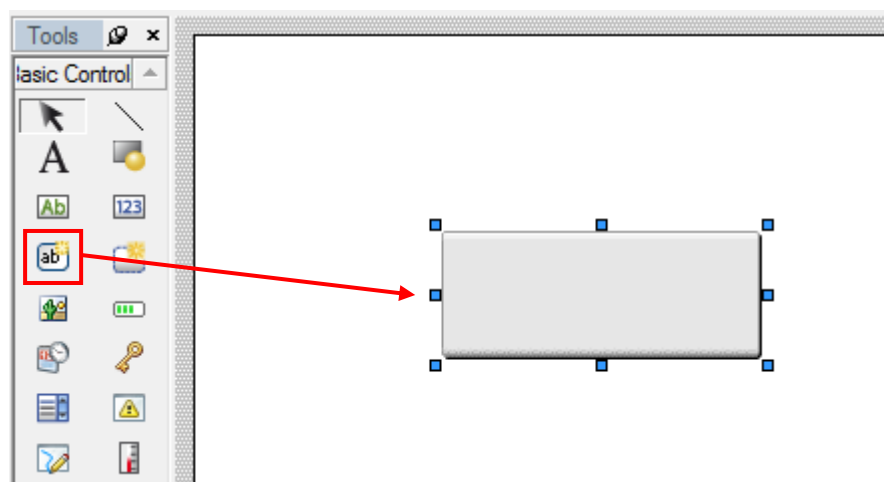
La part central de l'espai de treball es on es visualitza cada pantalla i on treballarem. Es la zona on crearem el disseny de les nostres pantalles i on es visualitzarà cada pantalla per poder configurar els nostres elements de treball, com ara gràfiques, imatges i estils de text i fonts.



Imatge 43: Propietats dels objectes de visualització

Una vegada familiaritzats amb tots els elements, objectes i les seves propietats ja es moment de començar el disseny de la nostra pantalla. Per això a continuació explicaré com a petit exemple, com crear un botó i assignar-li una acció per obrir una altre pantalla del nostre projecte.

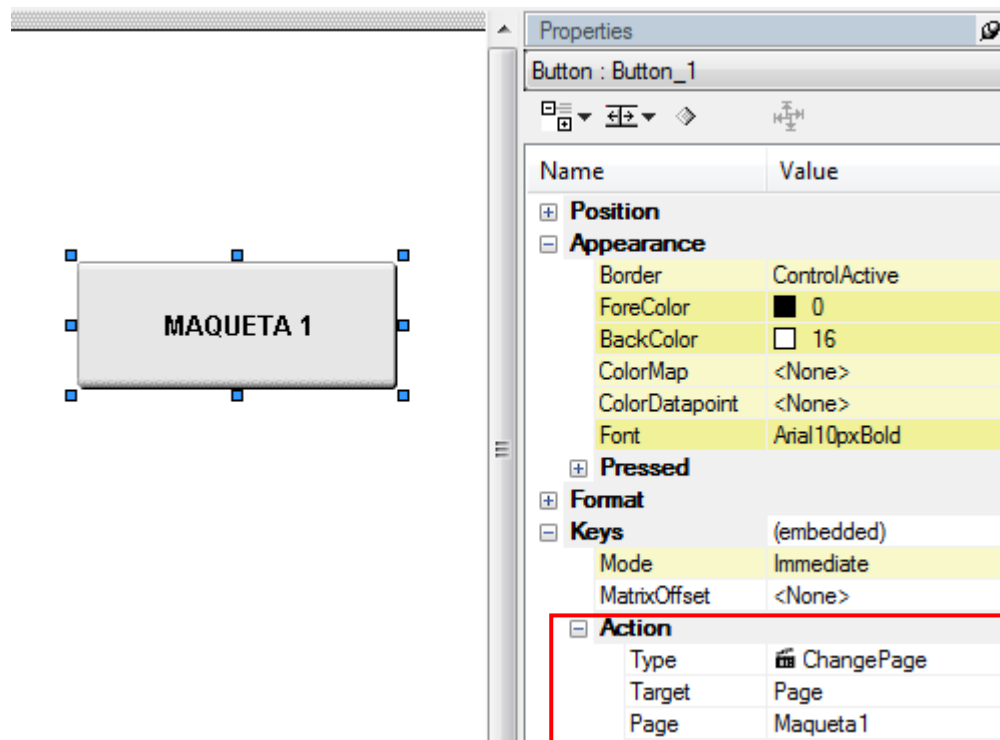
Per fer-ho només és tracta de fer clic sobre el símbol de botó de la finestra d'eines i dibuixar-lo sobre la pantalla central on tindrem oberta la pagina on vulguem col·locar-lo (veure Imatge 44).



Imatge 44: Afegir un botó a una pagina de visualització

Quan ja tinguem el nostre botó creat li assignem un nom per referir-nos a ell a través de la propietat *Name*. Podem escriure un text a mostrar a través de la propietat *Text*, podem canviar-li el color del text amb la propietat *ForeColor*, i la seva alineació amb el *Format*.

Finalment per fer que al pulsar el botó se'ns obri la pagina desitjada, anem a la propietat *Keys* i dintre d'aquesta a l'apartat *Action* escollim la opció *ChangePage* i li indiquem quina pagina volem que ens mostri (veure Imatge 45).

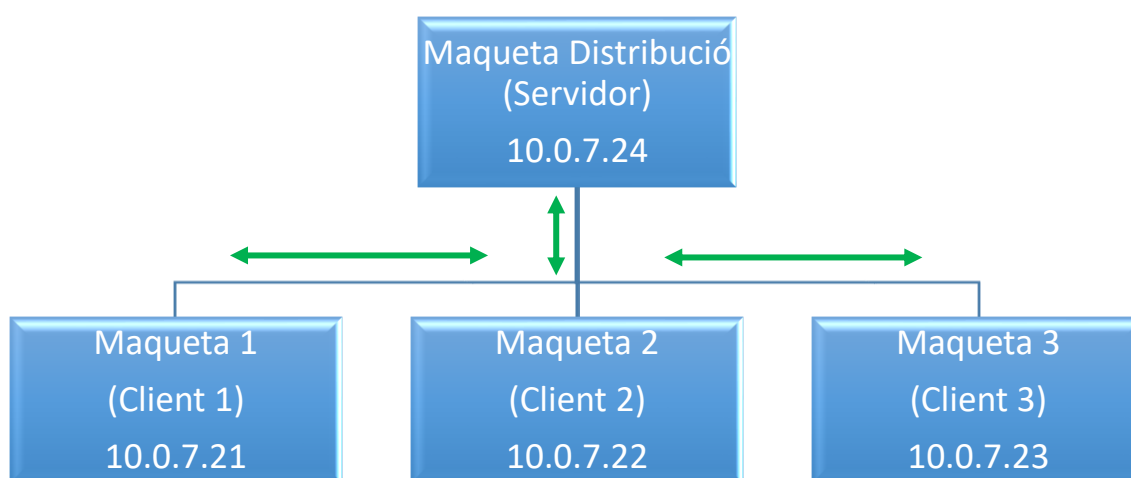


Imatge 45: Configuració botó en una visualització

### 3.1.3 Sistema de comunicació OPC UA

La última part i es podria dir que la més important del projecte, es la comunicació entre els 4 autòmats, ja que sense aquesta comunicació no seria possible que funcionés.

Tot el projecte esta basat en 4 PLC's com hem dit i s'ha estructurat, en un Servidor i 3 Clients. El servidor es el PLC de la maqueta de distribució i els clients son els PLC's de les maquetes 1, 2 i 3. El servidor es l'encarregat de demanar als clients les variables necessàries per actualitzar la pantalla de visualització i de proporcionar als clients les variables per engegar, parar i canviar de recepta en funció del que escollim a través de la pantalla.



Imatge 46: Diagrama de comunicació OPC UA

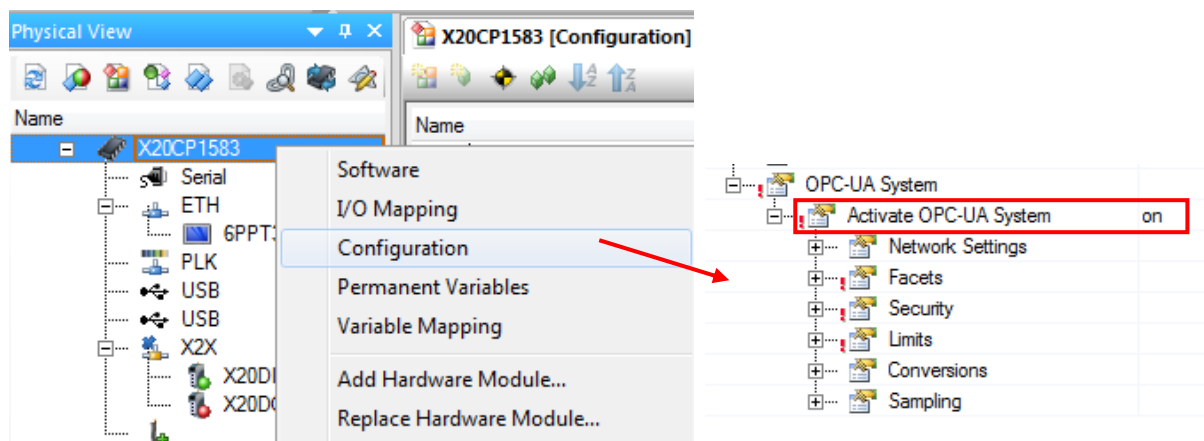
#### 3.1.3.1 Definició de OPC UA

L'OPC UA es basa en el principi client-servidor i permet una comunicació completa, des del sistema ERP fins als sensors i actuadors individuals, i permet la interoperabilitat semàntica als sistemes ciber-físics. OPC UA és flexible i totalment independent, i contribueix de manera decisiva a l'èxit de la quarta evolució industrial [1].

#### 3.1.3.2 Activació Sistema OPC UA

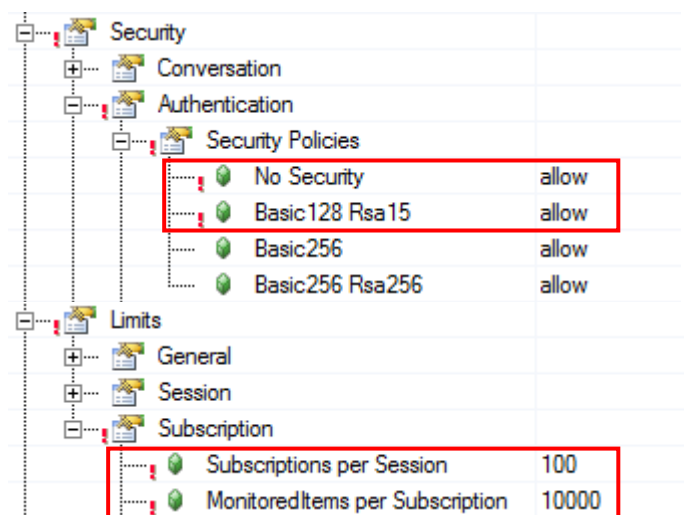
El primer pas que hem de fer per configurar el nostre PLC com a servidor o client, és activar el sistema OPC UA, per això hem d'anar a la pestanya de *Physical* i fer clic amb el botó dret sobre la CPU del nostre autòmat, i se'ns obrirà la finestra de configuració on podrem activar el sistema i modificar les seves propietats (veure Imatge 47).





Imatge 47: Activar Sistema OPC UA

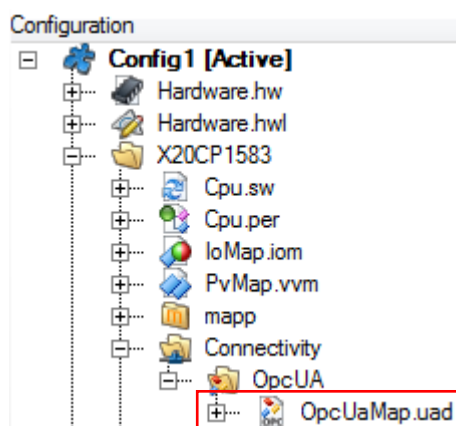
A l'apartat de xarxa, el numero de port el deixem tal i com està, que per defecte es el 4840. En el nostre cas només modifiquem els tipus d'accés permesos per el nostre servidor habilitant-los tots i en l'apartat dels límits, establint el nombre màxim de connexions per sessió a 100 i les variables monitoritzades per sessió a 10000 com es mostra a la Imatge 48.



Imatge 48: Configuració propietats sistema OPC UA

### 3.1.3.3 Configuració Sistema OPC UA

Per començar el primer que hem de fer és afegir un arxiu de configuració OPC UA dins de la nostra configuració del projecte. Ens dirigim a la pestanya de *Configuration View* i allà posant-nos sobre la carpeta de Connectivitat de la nostra CPU, busquem al catàleg d'objectes l'arxiu *OPC UA Default View File* i l'arrastrem cap a la carpeta OPC UA que penja de la carpeta anterior i ens apareixerà ja lligat l'arxiu a la nostra configuració.



Imatge 49: Arxiu de configuració OPC UA

Si l'obrim, ens trobem un arbre on hi ha totes les variables tant globals com locals del nostre projecte i per defecte apareixen totes des-habilitades, per habilitar les variables que vulguem compartir a través del OPC UA, només hem de fer doble clic sobre la variable que desitgem i a la finestra de propietats que se'ns obrirà, habilitar-la amb la opció *Enable* i la posem a *True*.

Un cop fet això ja tenim el nostre sistema OPC UA activat i configurat, amb les variables que vulguem també habilitades i ja podem passar a programar amb llenguatge escrit com s'ha de comportar tant el Servidor com els Clients.

### 3.1.3.4 Servidor OPC UA

Per poder utilitzar i programar les funcions proporcionades per el programar Automation Studio és necessari que estiguin incloses les llibreries *AsOpcUas* i *AsOpcUac* en el nostre projecte.

Per fer que un PLC treballi com a servidor OPC UA, no cal res més que el que hem fet fins ara, i és tant sols habilitar el sistema OPC i habilitar les variables com hem descrit anteriorment. Amb això n'hi haurà prou perquè un client es pugui connectar amb el servidor i escriure o llegir les variables.

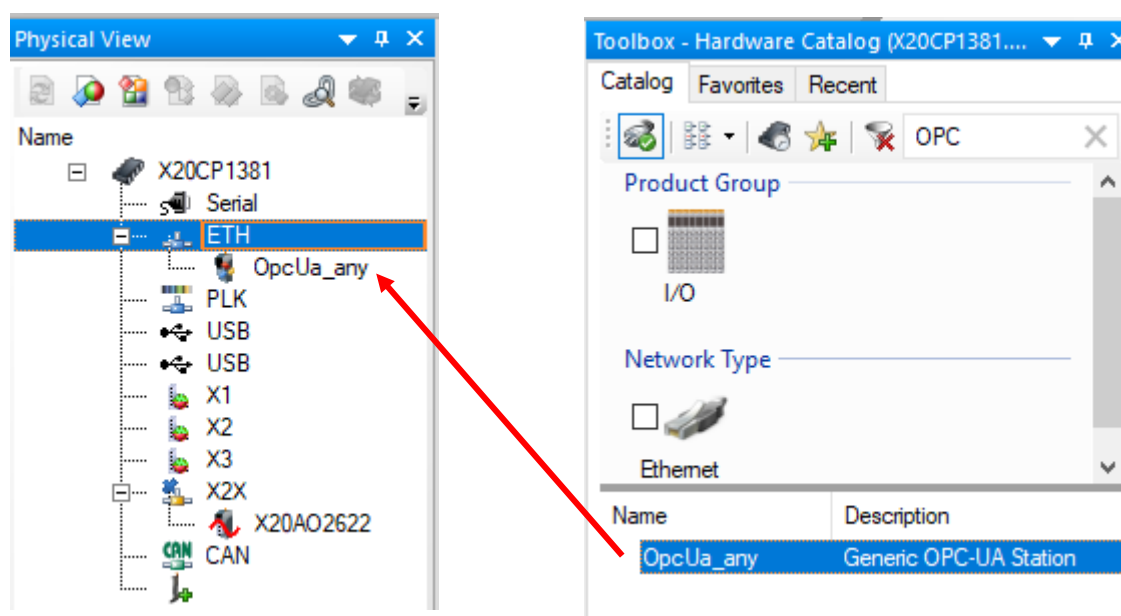
### 3.1.3.5 Client OPC UA

Seguidament es mostra i s'explica els passos a seguir per configurar els PLC's de les maquetes 1, 2 i 3 perquè treballin com a clients OPC UA [2].

Per poder utilitzar i programar les funcions proporcionades per el programar Automation Studio és necessari que estiguin incloses les llibreries *AsOpcUas* i *AsOpcUac* en el nostre projecte.

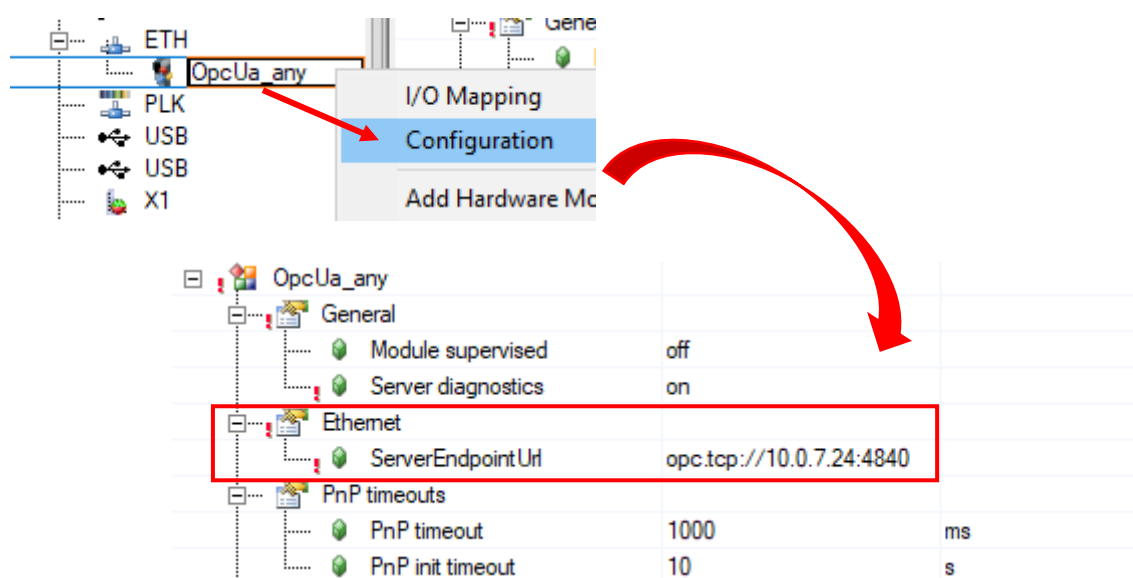
El primer pas després d'haver habilitat el sistema OPC UA i haver escollit quines son les variables que volem transmetre o permetre que el servidor hi accedeixi és el de configurar el hardware per permetre que el PLC es comporti com a client.

Per fer això ens dirigim a la pestanya *Physical* i ens col·loquem sobre del port de configuració Ethernet "ETH". A la finestra d'objectes situada a la dreta de la pantalla "Toolbox" introduïm la paraula OPC al buscador i ens apareixerà el component *OpcUa\_any*, aquest l'arrastrem fins al port Ethernet i ja podrem obrir-lo per configurar-lo (veure Imatge 50)



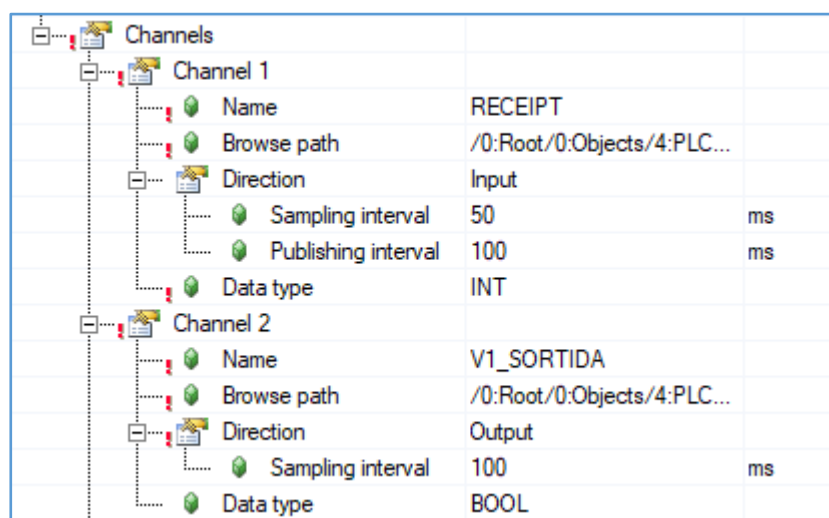
Imatge 50: Afegir hardware configuració OPCUa\_any

Una vegada afegit el dispositiu, hi ha dos passos a seguir per configurar-lo. El primer es obrir la configuració fent clic dret sobre ell i clicant a Configuració com es mostra en la Imatge 51, i se'ns obrirà la finestra on podrem definir la IP i el port del nostre servidor, en el nostre cas escriurem "opc.tcp://10.0.7.24:4840" sense les cometes.



Imatge 51: Configuració dispositiu OPCUa\_any

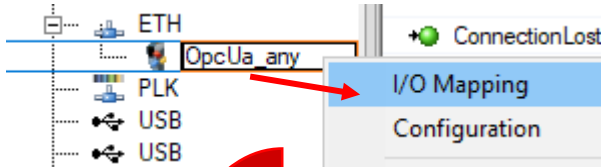
Aquí mateix es on s'hauran de crear i definir els nostres canals, un per cada variable i especificar si es d'entrada o de sortida i el tipus de variable que és. Seguidament es mostra un exemple d'un canal amb una variable d'entrada i un altre amb una de sortida.



Imatge 52: Canals d'entrada i sortida OPC UA

Per fer-ho, hem de definir un nom per el canal, escollir si es entrada o sortida, el tipus de variable (INT, BOOL, REAL...etc) i escriure la direcció on el client l'ha d'anar a buscar en el servidor per poder escriure-la o llegir-la. Aquesta direcció s'ha d'escriure de la següent manera: `"/0:Root/0:Objects/4:PLC/6:Modules/6:&:/6:Global PV/6:NOM DE LA VARIABLE"` sense les cometes i on posa *NOM DE LA VARIABLE* s'ha de substituir per el nom que té en el programa del servidor.

Per altre banda, un cop creats tots els canals, només queda assignar cada nom del canal amb la variable corresponent on volem que s'escrigui la dada o de on es llegeixi. Per assignar cada canal amb la variable corresponent s'ha de fer clic dret sobre el dispositiu OPCUa\_any i fer clic sobre *I/O Mapping* i se'ns obrirà la pantalla per poder assignar-los (veure Imatge 53).



+ RECEIPT	::RECEPTA	INT
+ V1_SORTIDA	::V1	BOOL
+ V2_SORTIDA	::V2	BOOL
+ V3_SORTIDA	::V3	BOOL
+ V4_SORTIDA	::V4	BOOL
+ LT1_SORTIDA	::LT1	BOOL
+ LT2_SORTIDA	::LT2	BOOL
+ LT3_SORTIDA	::LT3	BOOL
+ LT4_SORTIDA	::LT4	BOOL
+ LT5_SORTIDA	::LT5	BOOL
+ P1_SORTIDA	::P1	INT
+ LED_POWER	::POWER	BOOL
+ P_MARXA	::START_PANTALLA	BOOL
+ P_STOP	::STOP_PANTALLA	BOOL
+ INIT	::INICI	BOOL
+ RESET_IN	::RESET	BOOL
+ AM_OUT	::A_M	BOOL
+ V5_SORTIDA	::V5	INT
+ PT1_SORTIDA	::PT1	INT
+ LT6_SORTIDA	::LT6	INT
+ PID_KP	::PID_BOMBA_KP	REAL
+ PID_KI	::PID_BOMBA_TI	REAL
+ PID_KD	::PID_BOMBA_TD	REAL

Imatge 53: Assignació de variables al canal OPC UA

A la finestra que se'ns obre, podem veure com a la columna esquerra, ens apareixen els noms dels canals que hem creat anteriorment, i a la columna central es on hem d'escriure o buscar la variable dintre les variables globals o locals del nostre programa. El tipus de dada ja ens ve donada per el tipus que hem pre-assignat al crear a canal, i al buscar la variable a assignar, ens serà mes fàcil ja que el programa només ens mostrarà les variables que siguin del mateix tipus.

Fet tot això ja tenim configurat tant els clients com el servidor per poder enviar i rebre totes les senyals i variables dels nostres programes a través de la comunicació OPC UA.

## 4. METODOLOGIA

En aquest apartat és on ja entrem en profunditat en com s'ha resolt el problema inicial del projecte i on veurem els diferents programes, la manera com s'ha resolt la comunicació entre els 4 PLC's i el disseny de la pantalla tàctil.

L'estratègia que s'ha seguit es la de crear un programa igual per cada un dels PLC's de les maquetes base que per tant treballaran igual i al mateix moment, i un programa diferent per el PLC de la maqueta de distribució. La base d'aquest projecte es centra en treballar amb receptes, aquestes receptes es triaran des de la pantalla tàctil a través de diferents botons i seran enviades a través de la comunicació OPC UA des del servidor fins a cada client, per així poder activar o desactivar cada programa prèviament creat a partir de diferents Grafcets, un per cada recepta.

En total s'han creat 6 receptes, aquestes s'han distribuït de la següent manera:

Les 3 primeres són iguals per cada maqueta i s'executen a la vegada a les 3 maquetes base, i les 3 ultimes són governades a través de la maqueta de distribució i son les que serveixen per enviar aigua des del dipòsit de cada maqueta en concret, en funció del numero de recepta (recepta 4 – maqueta 1, recepta 5 – maqueta 2 i recepta 6 – maqueta 3) fins al dipòsit de la maqueta central de distribució.

Paral·lelament, cada PLC te el seu programa per realitzar la comunicació entre autòmats, el programa per els clients i per el servidor, a les maquetes 1, 2 i 3 i a la de distribució respectivament. Aquesta comunicació serveix com hem dit anteriorment per enviar el numero de recepta que volem, per enviar a la pantalla les senyals necessàries per visualitzar l'estat actual de cada maqueta i per rebre de la mateixa les accions a realitzar en cada moment.

### 4.1 Programació maquetes base

A continuació es descriu, el diagrama de Gemma on s'han definit els diferents modes de funcionament i es mostren els seus Grafcets, després veurem una per una les 3 receptes per les maquetes base i es mostraran els diferents Grafcets per cada una d'elles.

#### 4.1.1 Diagrama GEMMA maquetes base

Per controlar els modes de funcionament de les maquetes 1, 2 i 3 s'ha dissenyat el següent diagrama igual per cada una d'elles, no es únic per totes 3, sinó que cada programa de cada maqueta tindrà el seu diagrama. En aquest projecte només s'ha tingut en compte dos modes de funcionament, el mode automàtic i el mode manual.

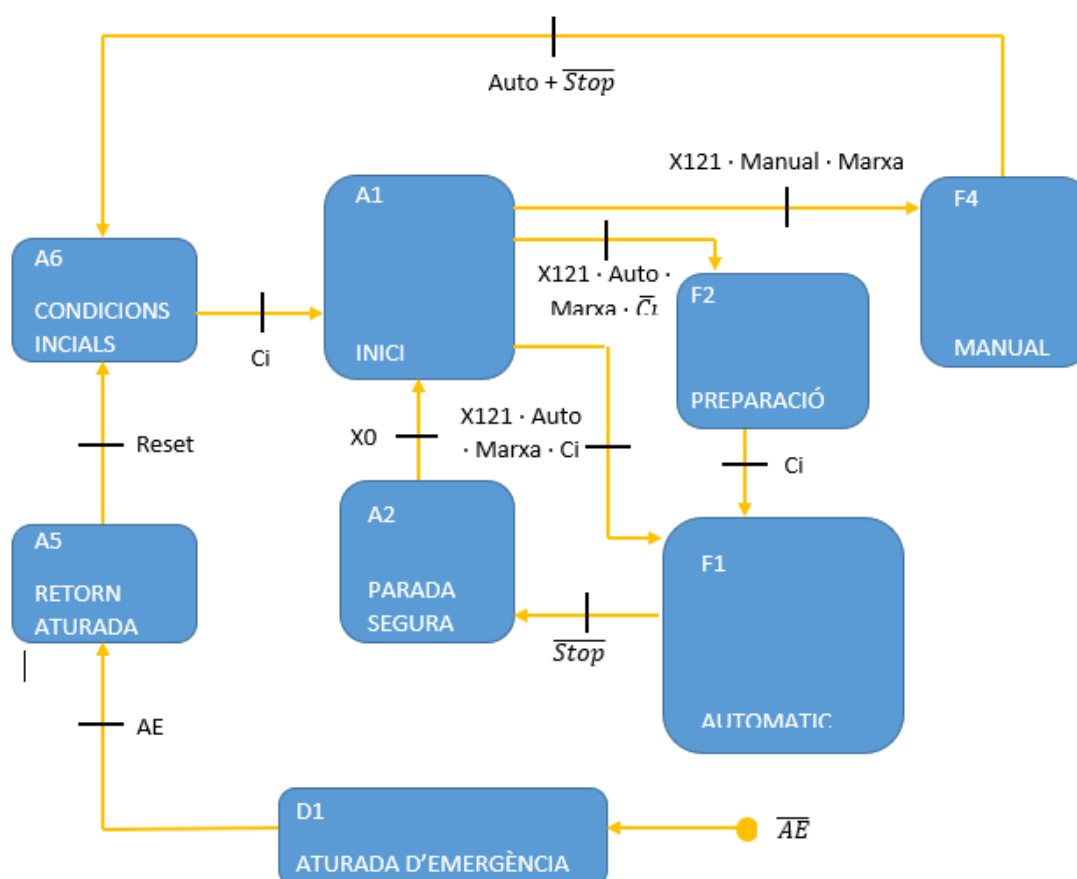
Per entrar al mode automàtic o manual es necessari que estiguem a l'etapa 121 (Etapa d'espera mentre no hi hagi el paro d'emergència polsat) i a més a més seleccionem un dels dos modes i polsem Marxa, depenent de quin es seleccioni s'entrarà a un mode o a un altre.

Si abans d'entrar al mode automàtic no es compleixen les condicions inicials establertes prèviament (es descriuen a continuació) es passarà per un estat previ, on es posarà el sistema en condicions inicials per poder iniciar el programa.

Per altre banda, per sortir del mode automàtic de forma segura es necessari polsar el polsador de stop i que a més a més estiguem a l'etapa inicial, i per sortir del mode manual n'hi ha prou en posar el selector en automàtic o en polsar l'stop.

Si en algun moment donat es prem el polsador d'emergència, automàticament es parará tot el programa i s'entrarà en mode d'aturada d'emergència, que per sortir-ne farà falta treure el paro d'emergència i un cop fet el reset, passarem al mode de condicions inicials, que si aquestes son certes directament ens tornarem a trobar a l'inici.

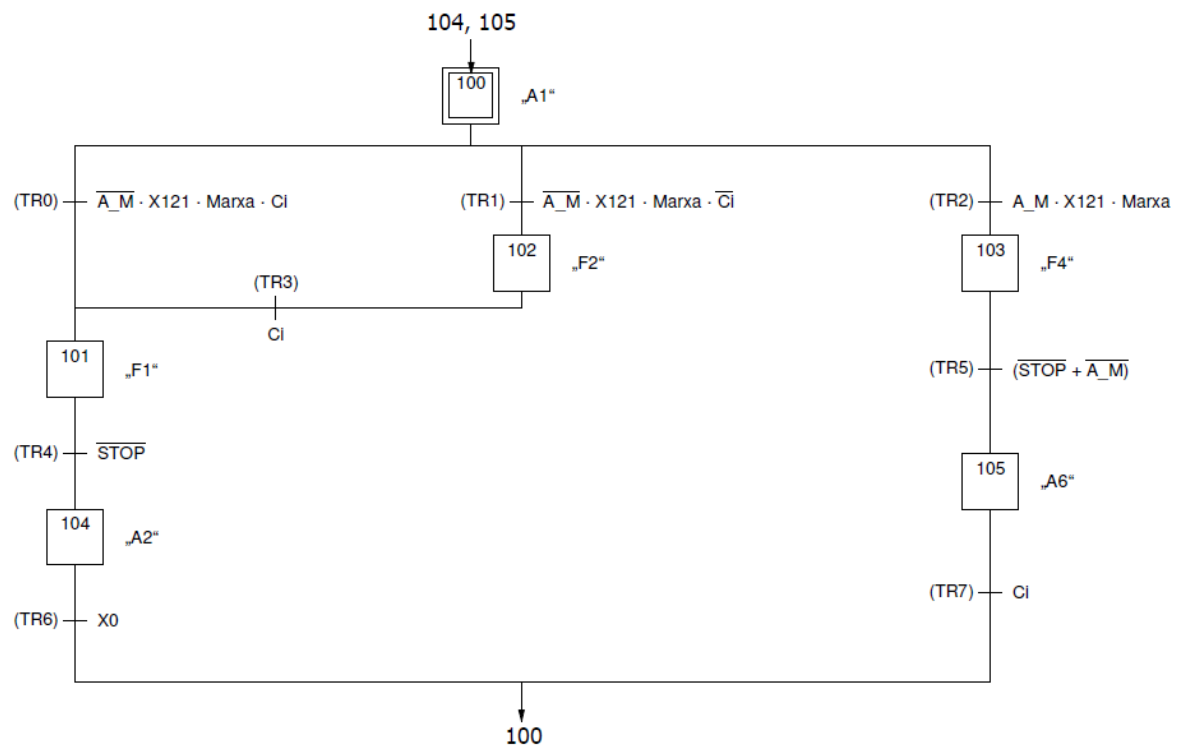
Les condicions inicials que s'han decidit, són que no hi hagi nivell al tanc superior de la maqueta.



Imatge 54: Diagrama GEMMA maqueta base

#### 4.1.1.1 Grafcet diagrama GEMMA maquetes base

El següent Grafcet es la representació del diagrama GEMMA igual per cada una de les maquetes base, on veiem que les transicions i etapes mostrades anteriorment al diagrama.



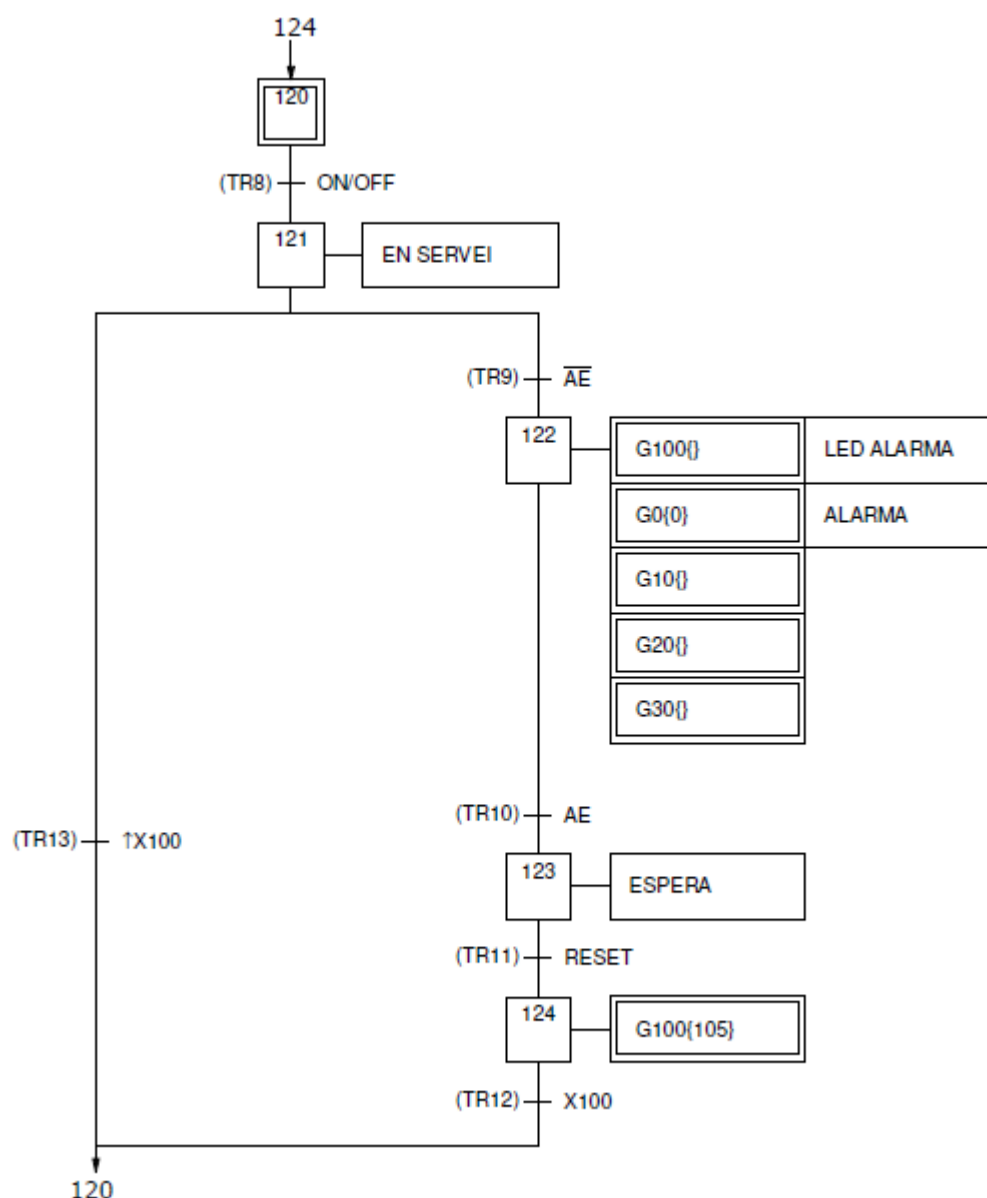
Imatge 55: Grafcet 100, Diagrama GEMMA maquetes base



#### 4.1.1.2 Grafcet parada d'emergència maquetes base

A continuació es mostra el Grafcet de l'aturada d'emergència, on podem veure que mentre no estigui polsat el paro d'emergència, el Grafcet es manté a l'etapa 121 que es la que ens permet poder iniciar el programa principal per escollir la recepta.

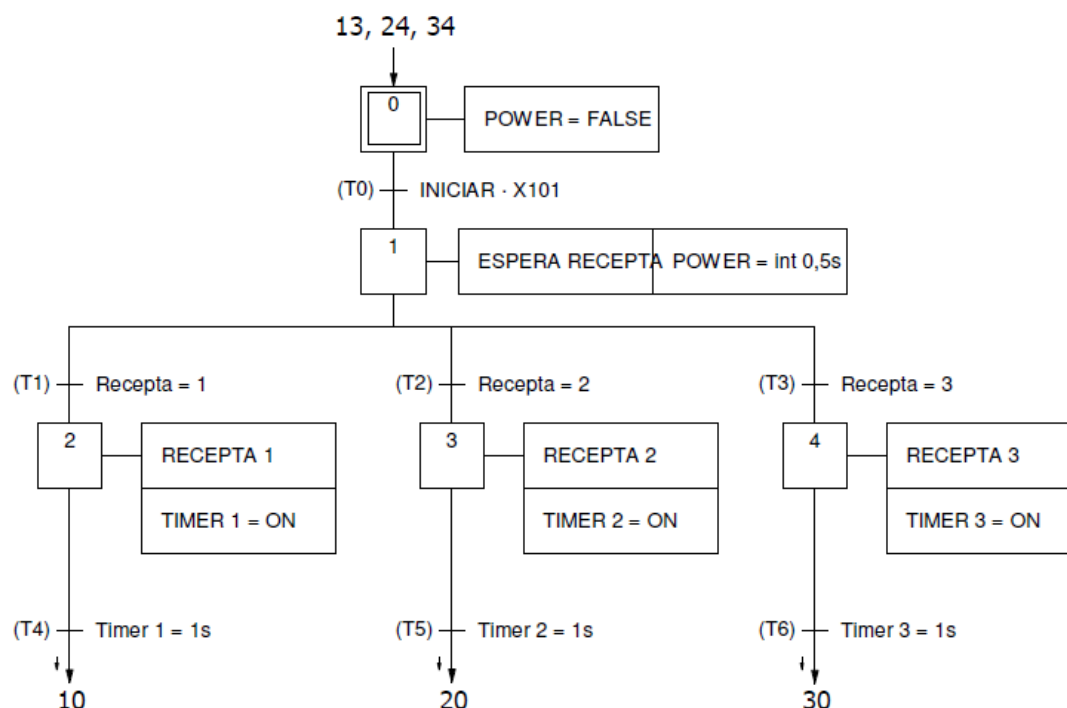
Si en algun moment es polsa el paro, entrarem a l'etapa 122 i es aquí on es forçarà el programa a l'etapa 0 a l'espera que el Grafcet 100 (Diagrama Gemma) es torni a iniciar.



Imatge 56: Grafcet 120, parada d'emergència maquetes base

#### 4.1.2 Espera de recepta maquetes base

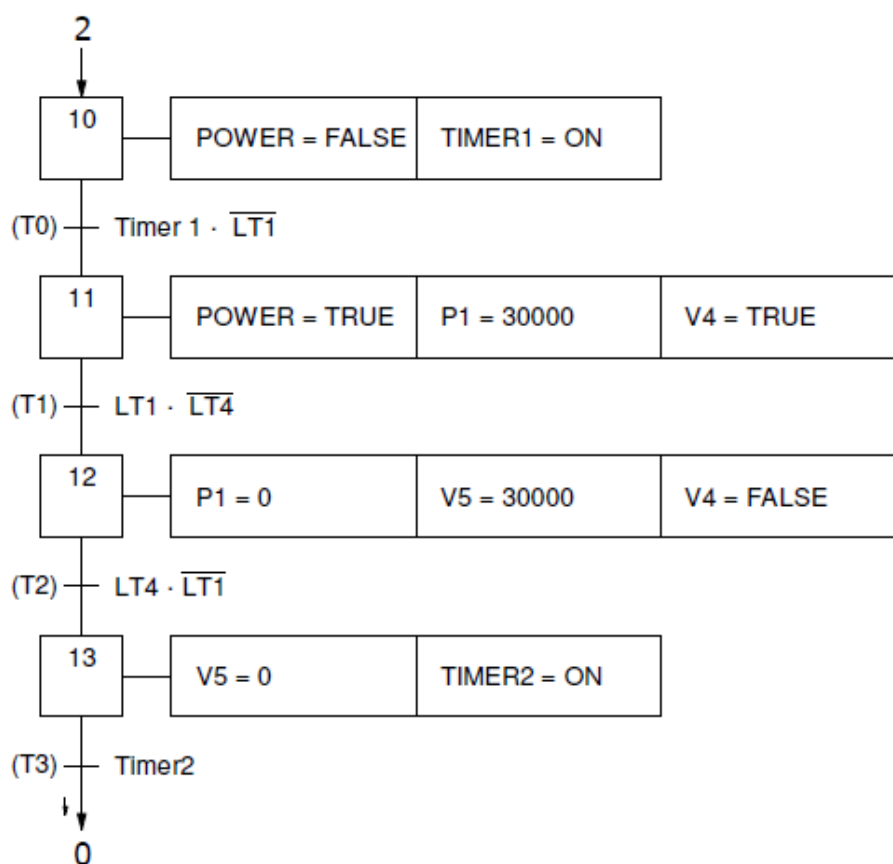
Un cop iniciades les maquetes, aquestes queden a l'espera de rebre un numero de recepta per començar a funcionar, per identificar que estan esperant una recepta, el LED de POWER es queda en intermitència i a la pantalla tàtil ens surt el missatge de "Esperant recepta". Un cop indicat el numero de recepta (1, 2 o 3) el programa de cada maqueta entra en un estat o un altre i s'executa el programa de la recepta en concret i totes 3 a la vegada.



Imatge 57: Grafcet 0, espera de recepta maquetes base

#### 4.1.3 Recepta 1 maquetes base

Si la recepta escollida es la numero 1, com podem veure a la Imatge 57 entrarem a l'etapa 10. Aquesta recepta el que fa es que primer es comprova que el dipòsit superior estigui buit, si es així s'obre la vàlvula V4 i s'activa la bomba juntament amb el LED de POWER per indicar que la maqueta esta funcionant. La bomba seguirà activa fins que s'assoleixi el nivell baix en el tanc superior i el nivell del tanc inferior estigui per sota de la meitat. Un cop es compleixi això, es para la bomba i s'obra la vàlvula V5 per buidar el dipòsit i retornar l'aigua al tanc inferior on estava inicialment fins que el nivell del tanc inferior torni a estar a la meitat (LT4) i el tanc superior estigui buit. Quan tinguem el tanc superior buit, es tanca la vàlvula V5 i es tornarà a l'etapa 0, per tornar a repetir aquesta recepta o començar-ne una de nova.



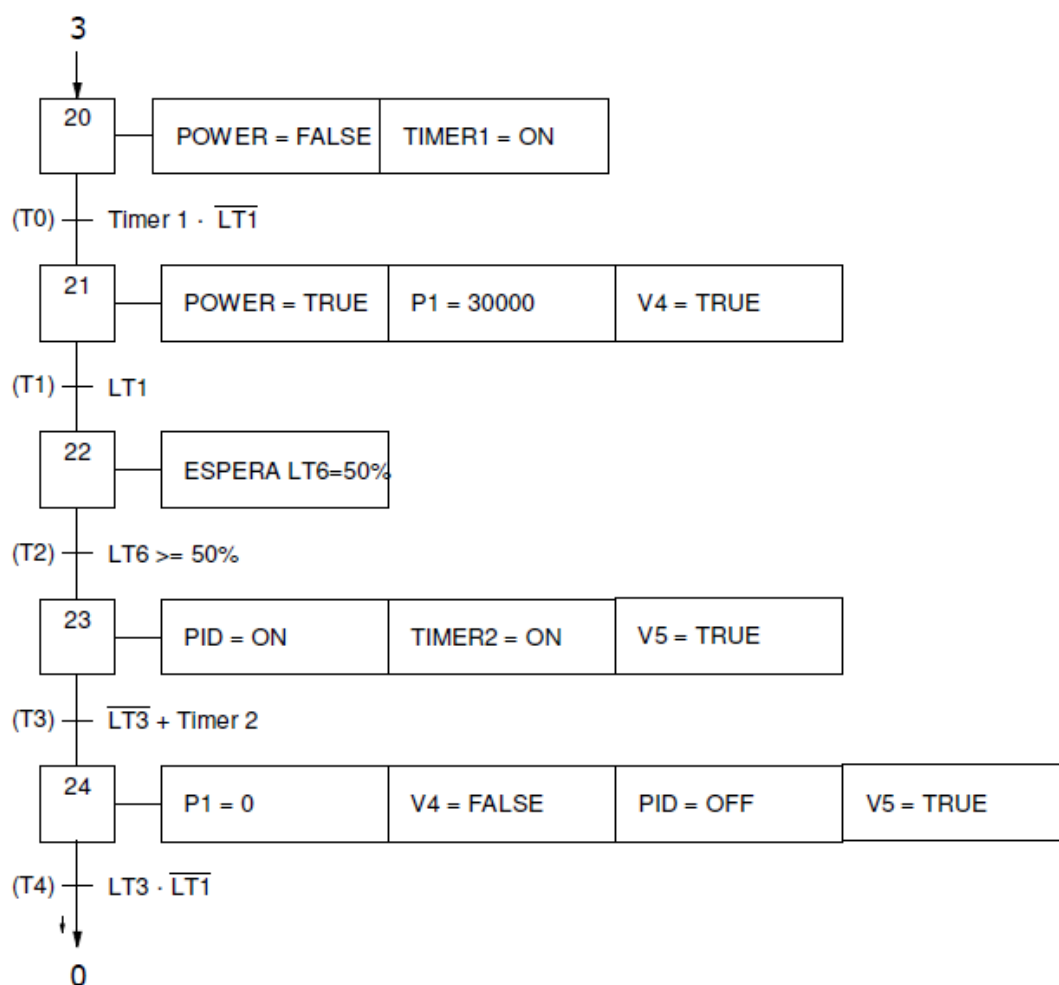
Imatge 58: Grafset 10, Recepta 1 maquetes base

#### 4.1.4 Recepta 2 maquetes base

Si la recepta escollida es la 2, entrarem a l'etapa 20.

A la recepta 2, el programa comença igual que la recepta 1, comprovant que no hi hagi aigua al tanc alt i un cop comprovat s'activa la bomba i s'obre la vàlvula V4 juntament amb el LED de POWER per indicar que la maqueta està funcionant. Un cop assolit el nivell mínim al tanc superior, aquí es on entra la diferència amb la recepta 1 i es on comença a funcionar el control PID per regular la bomba per mantenir el nivell en el dipòsit alt al 50%, llegit a través del sensor de nivell LT6, tot això obrint la vàlvula V5 al 100%.

El control PID serà l'encarregat de mantenir el nivell en el tanc superior sempre al 50% durant un temps preestablert de 1 minut mentrestant la vàlvula seguirà oberta al 100% sempre i quan hi hagi nivell mínim al tanc baix, si abans que passi el temps per el motiu que fos el tanc inferior es quedés sense aigua es parará la bomba, la vàlvula V4, el control PID i es mantindrà la vàlvula V5 oberta per recuperar el nivell. Un cop recuperat el nivell mínim i el tanc superior buit, es tornarà a l'etapa 0 per tornar a començar aquesta recepta o la que s'hagi escollit.



Imatge 59: Grafset 20, recepta 2 maquetes base

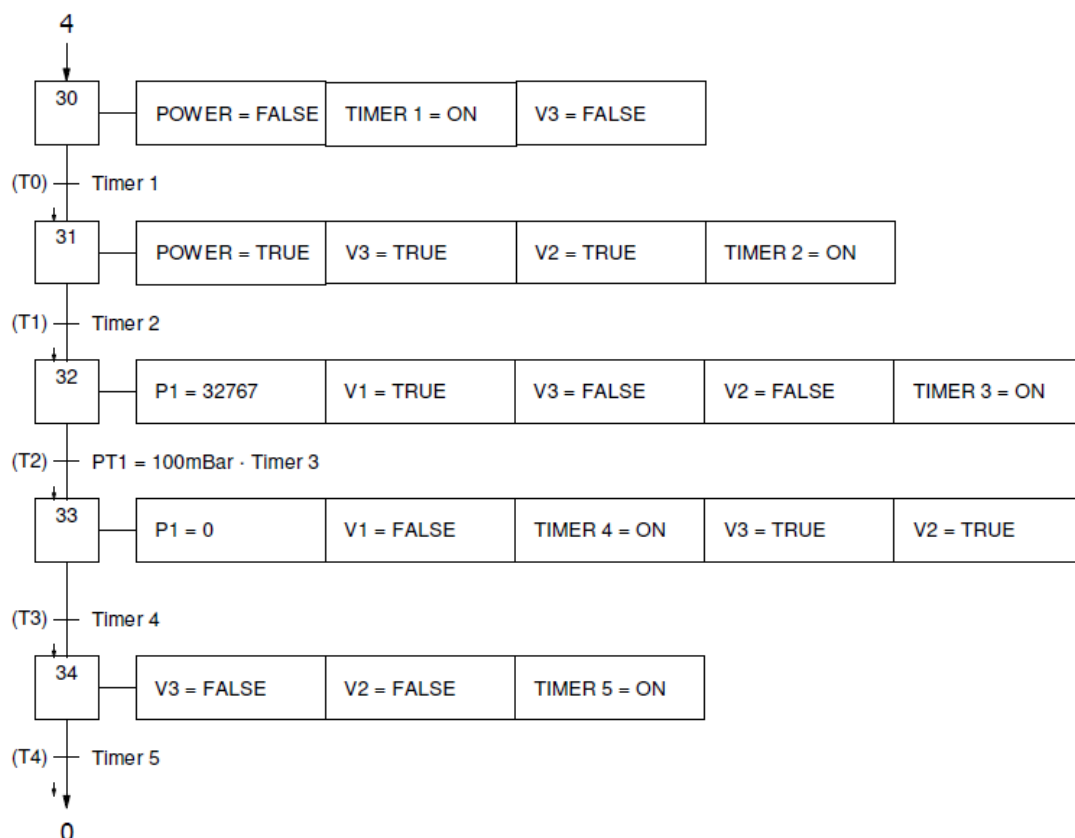
#### 4.1.5 Recepta 3 maquetes base

Per últim, si l'escollida es la recepta 3 el programa passarà a l'etapa 30.

En aquesta última recepta el programa pensat és que s'obren les vàlvules V2 i V3 durant 20 segons perquè el nivell entre el tanc pressuritzat i el nivell d'aigua en el tanc inferior de la maqueta s'igualin. Un cop igualats els nivells, es tanca la vàlvula V2 i es manté oberta la V3 mentre s'obra al mateix moment la vàlvula V1 i s'engega la bomba per crear una pressió dintre el vas pressuritzat.

Una vegada assolida la pressió desitjada de 100mBar i passats 5 segons, es para la bomba i es tanca la vàlvula V1, per tornar a obrir les vàlvules V2 i V3 i així buidar el vas pressuritzat i tornar l'aigua al dipòsit inferior de la maqueta.

Un cop acabada aquesta recepta, es retorna a l'etapa 0 per tornar a iniciar aquesta recepta o una diferent si s'ha escollit un numero diferent.



Imatge 60: Grafcet 30, recepta 3 maquetes base

## 4.2 Programació maqueta distribució

A continuació es descriu, el diagrama de Gemma de la maqueta de distribució on s'han definit els diferents modes de funcionament i es mostren els seus Grafcets, després veurem una per una les 3 receptes per la maqueta de distribució i es mostraran els diferents Grafcets per cada una d'elles.

Les tres receptes de la maqueta tenen el mateix objectiu, primer es comprova quin dipòsit es el que té menys aigua (recepta 4 – maqueta 1, recepta 5 – maqueta 2, recepta 6 – maqueta 3) si es el tanc inferior de la maqueta base, es passa l'aigua del tanc inferior cap al dipòsit de la maqueta central, i un cop assolit el nivell, es retorna l'aigua cap a la maqueta base (1, 2 o 3).

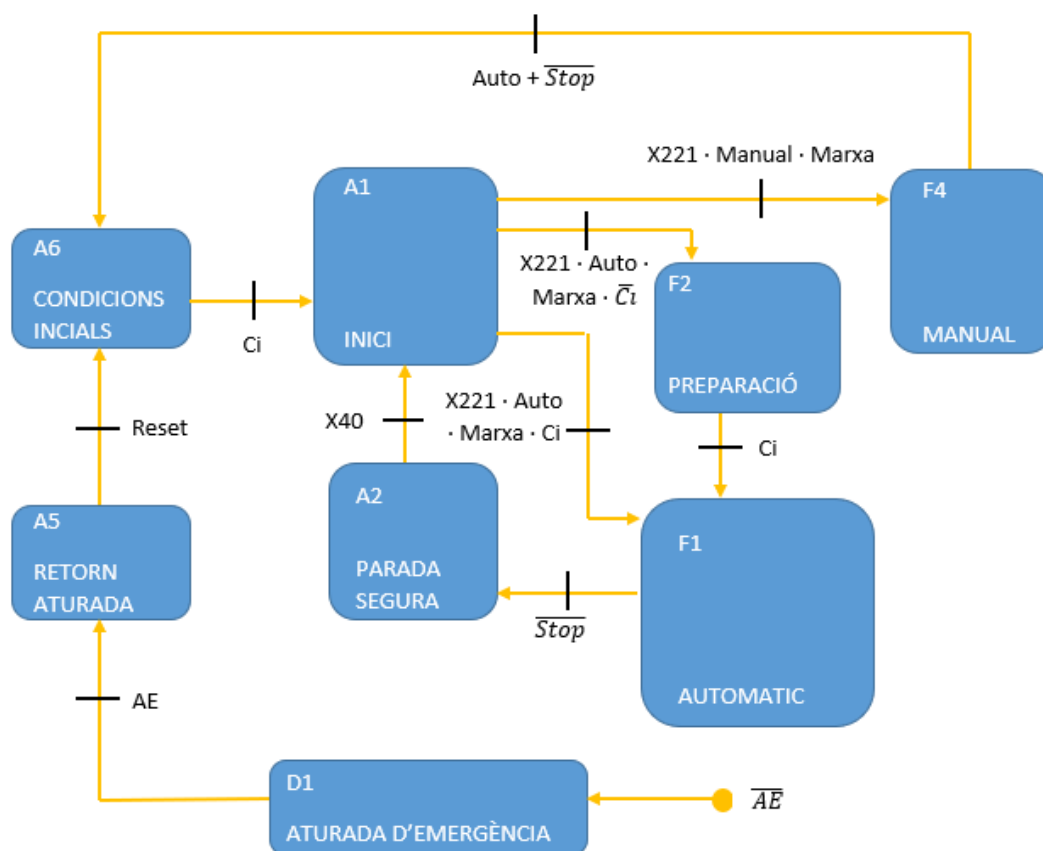
En canvi si es el dipòsit de la maqueta de distribució la que te menys aigua, s'envia aigua cap a la maqueta base i després es retorna l'aigua de nou cap al dipòsit de la maqueta de distribució.

#### 4.2.1 Diagrama GEMMA maqueta distribució

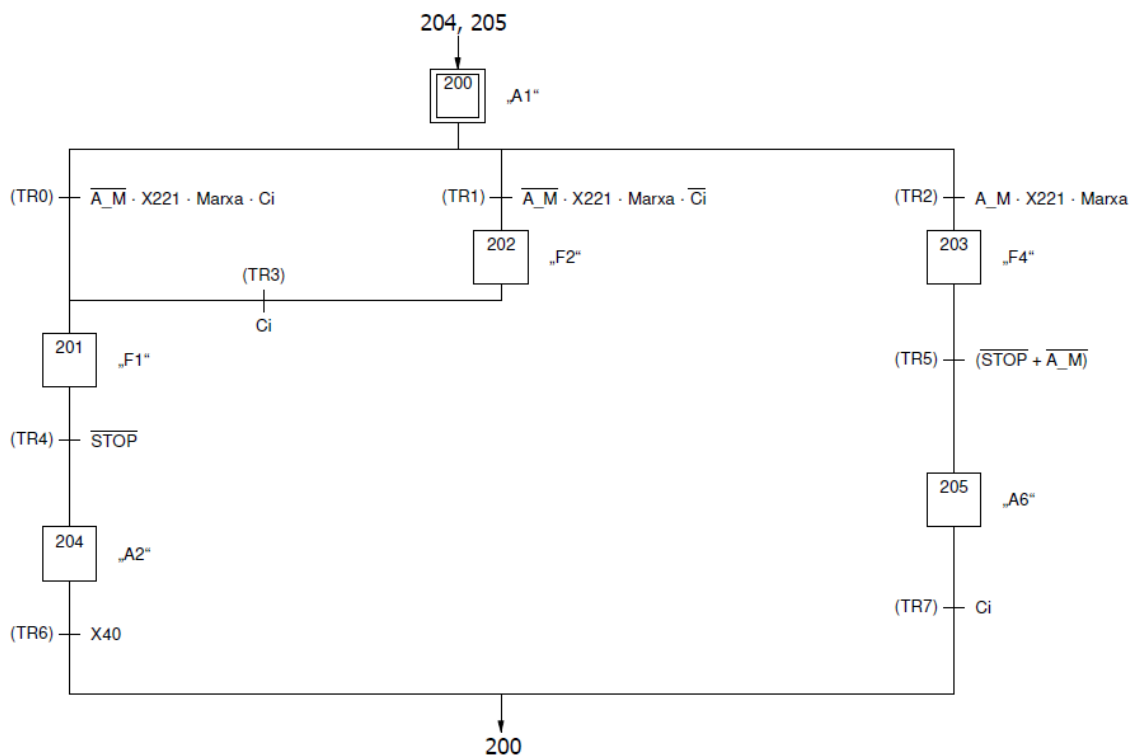
Per controlar els modes de funcionament de la maqueta de distribució s'ha dissenyat el següent diagrama. En aquest projecte només s'ha tingut en compte dos modes de funcionament, el mode automàtic i el mode manual.

Per entrar al mode automàtic o manual es necessari que estiguem a l'etapa 221 (Etapa d'espera mentre no hi hagi el paro d'emergència polsat) i a més a més seleccionem un dels dos modes i polsem Marxa, depenent de quin es seleccioni s'entrarà a un mode o a un altre. Si abans d'entrar al mode automàtic no es compleixen les condicions inicials establertes prèviament (es descriuen a continuació) es passarà per un estat previ, on es posarà el sistema en condicions inicials per poder iniciar el programa. Aquestes condicions inicials, són que cap dels dos tancs estigui buit.

Per altre banda, per sortir del mode automàtic de forma segura es necessari polsar el polsador de stop i que a més a més estiguem a l'etapa inicial, i per sortir del mode manual n'hi ha prou en posar el selector en automàtic o en polsar l'stop. Si en algun moment donat es prem el polsador d'emergència, automàticament es parará tot el programa i s'entrarà en mode d'aturada d'emergència, que per sortir-ne farà falta treure el paro d'emergència i un cop fet el reset, passarem al mode de condicions inicials, que si aquestes son certes directament ens tornarem a trobar a l'inici.



Imatge 61: Diagrama GEMMA maqueta distribució

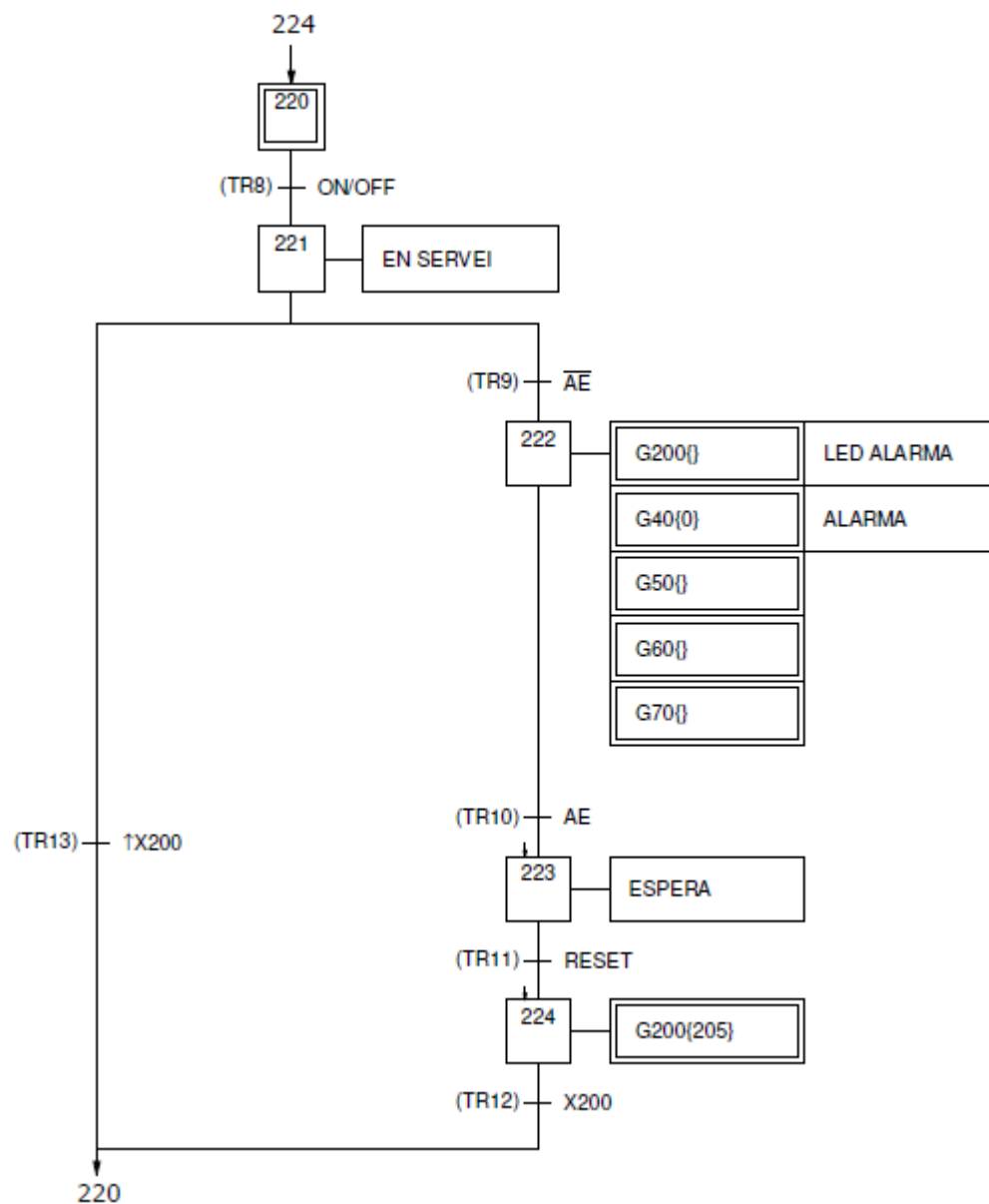


Imatge 62: Grafcet 200, diagrama GEMMA maqueta distribució

#### 4.2.1.2 Grafcet parada d'emergència maqueta distribució

A continuació es mostra el Grafcet de l'aturada d'emergència de la maqueta de distribució, on podem veure que mentre no estigui polsat el paro d'emergència, el Grafcet es manté a l'etapa 221 que es la que ens permet poder iniciar el programa principal per escollir la recepta.

Si en algun moment es polsa el paro, entrarem a l'etapa 222 i es aquí on es forçarà el programa a l'etapa 40 a l'espera que el Grafcet 200 (Diagrama Gemma) es torni a iniciar.

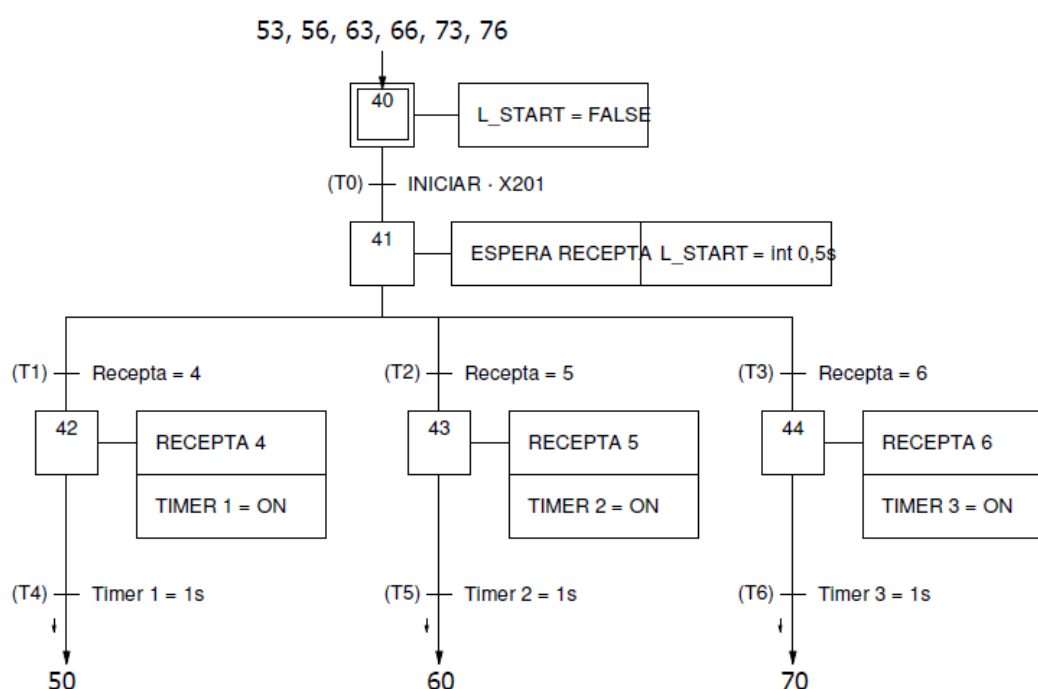


Imatge 63: Grafcet 220, parada d'emergència maqueta distribució



#### 4.2.2 Espera de recepta maqueta distribució

Un cop iniciada la maqueta, aquesta queda a l'espera de rebre un numero de recepta per començar a funcionar, per identificar que està esperant una recepta, el LED de POWER es queda en intermitència i a la pantalla tàctil ens surt el missatge de "Esperant recepta". Un cop indicat el numero de recepta (4, 5 o 6) el programa de la maqueta entra en un estat o un altre i s'executa el programa de la recepta en concret.



Imatge 64: Grafet 40, espera de recepta maqueta distribució

#### 4.2.3 Recepta 4 maqueta distribució

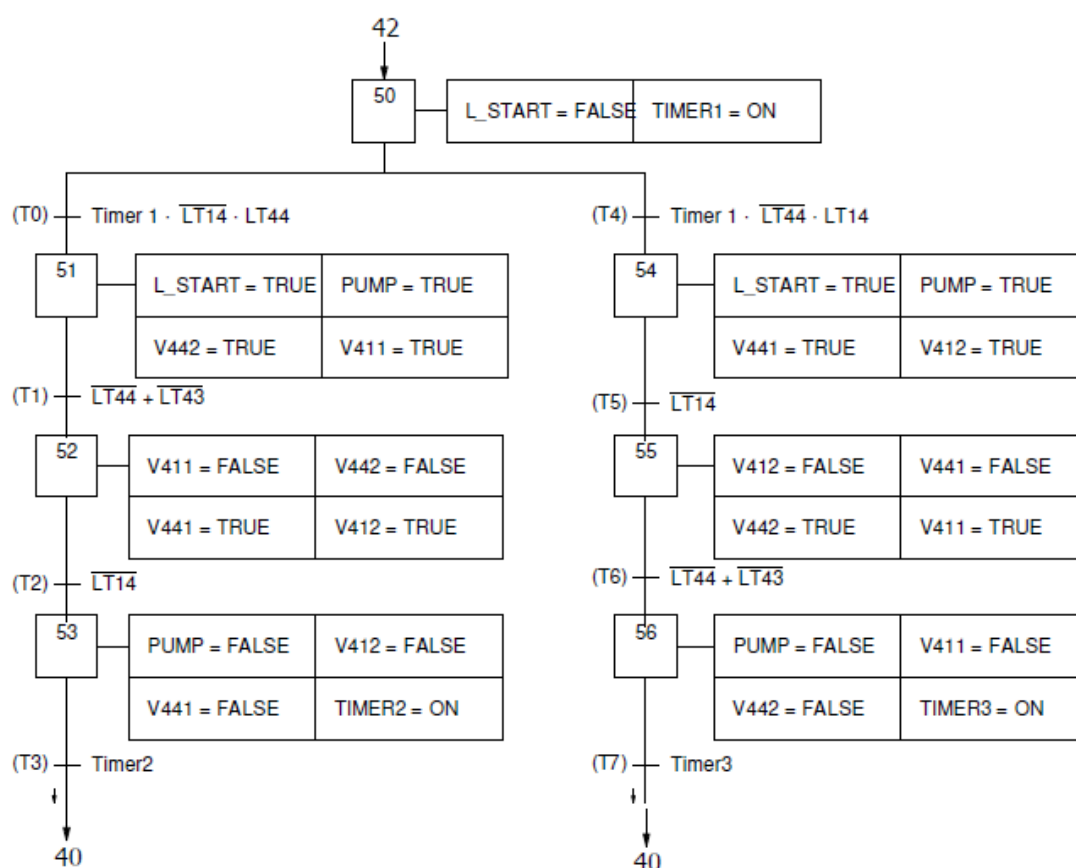
La recepta numero 4, té com a principi de funcionament comprovar quin dipòsit entre el tanc inferior de la maqueta 1 o el tanc de la maqueta de distribució es el que té menys aigua.

Si el dipòsit amb menys aigua es el de la maqueta numero 1, s'obren les vàlvules V442 i V411 i s'engega la bomba per bombejar aigua del tanc de la maqueta central cap al tanc inferior de la maqueta 1. Un cop tenim nivell mínim al tanc de la maqueta 1, es tanquen aquestes vàlvules i s'obren la V441 i V412 mantenint la bomba engegada i així tornant l'aigua cap al tanc de la maqueta de distribució.

Quan el nivell del tanc inferior de la maqueta 1 torna a estar com estava inicialment, es tanquen totes les vàlvules i es para la bomba i es retorna el programa a l'estat 40 esperant una nova recepta.

Si per el contrari el dipòsit amb menys aigua es el de la maqueta de distribució, s'obren les vàlvules V441 i V412 i s'engega la bomba i es bombeja aigua cap al tanc de la maqueta central fins que el tanc de la maqueta 1 perd el nivell mínim. Un cop passa això, es tanquen aquestes vàlvules i s'obren les vàlvules V442 i V411 per retornar l'aigua cap al tanc inferior, fins que el tanc central torna a quedar sense nivell mínim, es llavors quan es tanquen totes les vàlvules i es para la bomba i es retorna el programa a l'etapa 40.

El programa amb llenguatge escrit de la recepta 4 es pot veure a l'annex Programa Recepta 4.



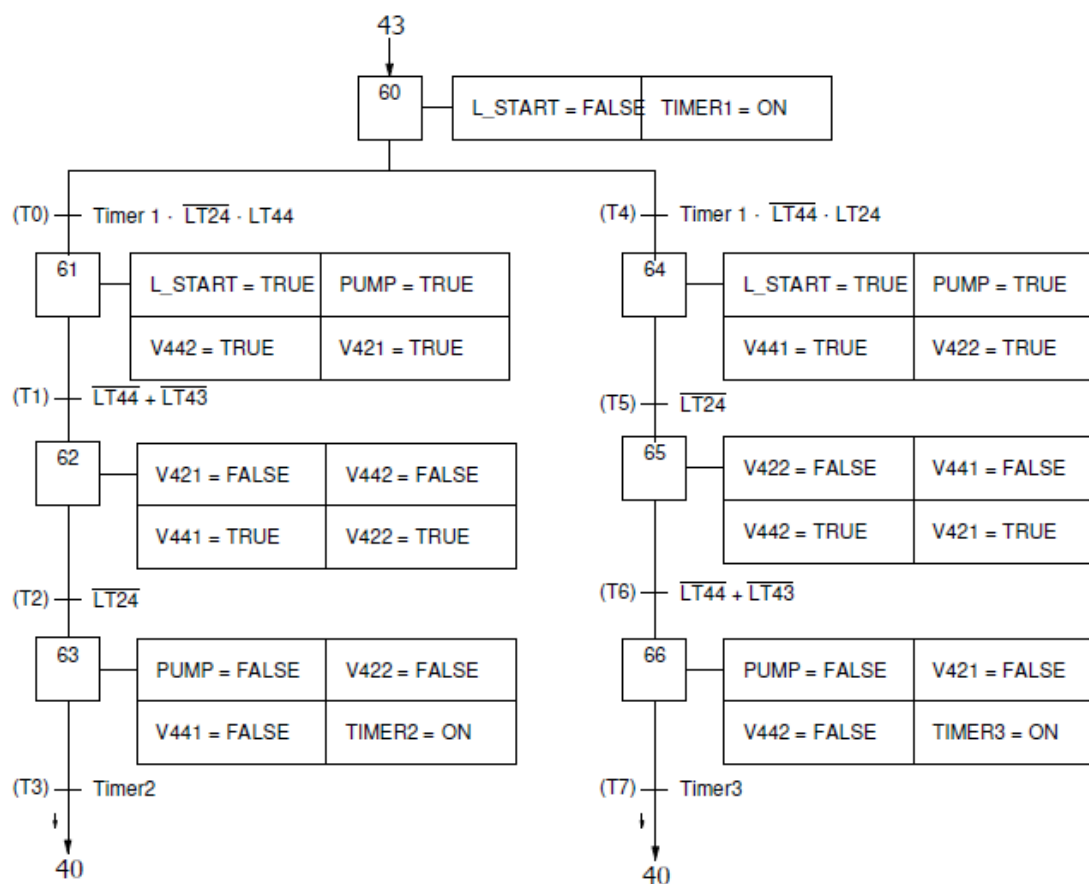
Imatge 65: Grafcet 50, recepta 4 maqueta distribució

#### 4.2.4 Recepta 5 maqueta distribució

Si la recepta escollida es la 5, entrarem a l'etapa 60.

A la recepta 5, el programa es comporta exactament igual que la recepta 4, però amb l'única diferència que en lloc de comparar nivells i traspasar aigua de la maqueta de distribució cap a la maqueta 1 o viceversa, en aquest cas es entre la maqueta 2 i la maqueta de distribució.

El programa amb llenguatge escrit de la recepta 5 es pot veure a l'annex Programa Recepta 5.



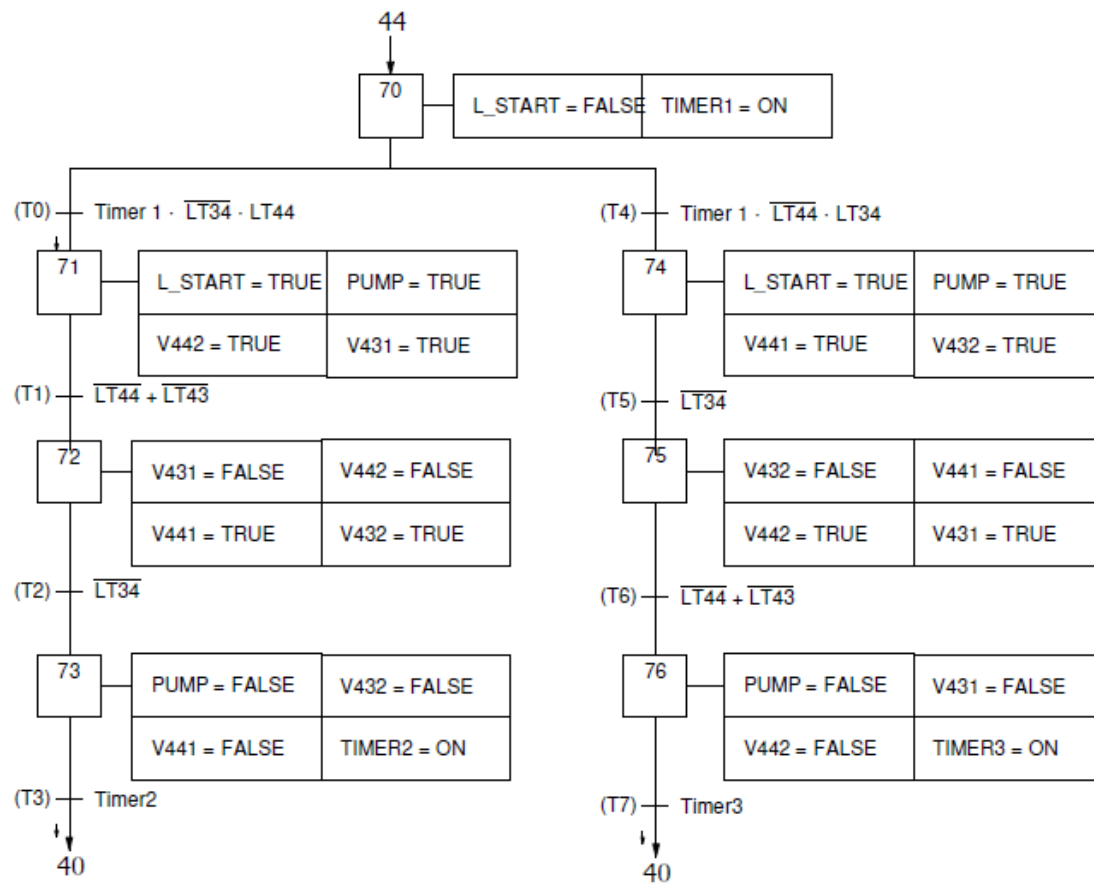
Imatge 66: Grafcet 60, recepta 5 maqueta distribució

#### 4.2.5 Recepta 6 maqueta distribució

Per últim, si l'escollida es la recepta 6 el programa passarà a l'etapa 70.

Aquesta ultima, igual que les dues receptes anteriors, està programada igual però també amb la diferència que en aquest cas, les maquetes que es traspassen l'aigua l'una amb l'altre, son la maqueta numero 3 i la maqueta de distribució, però amb el mateix principi de funcionament.

El programa amb llenguatge escrit de la recepta 6 es pot veure a l'annex Programa Recepta 6.

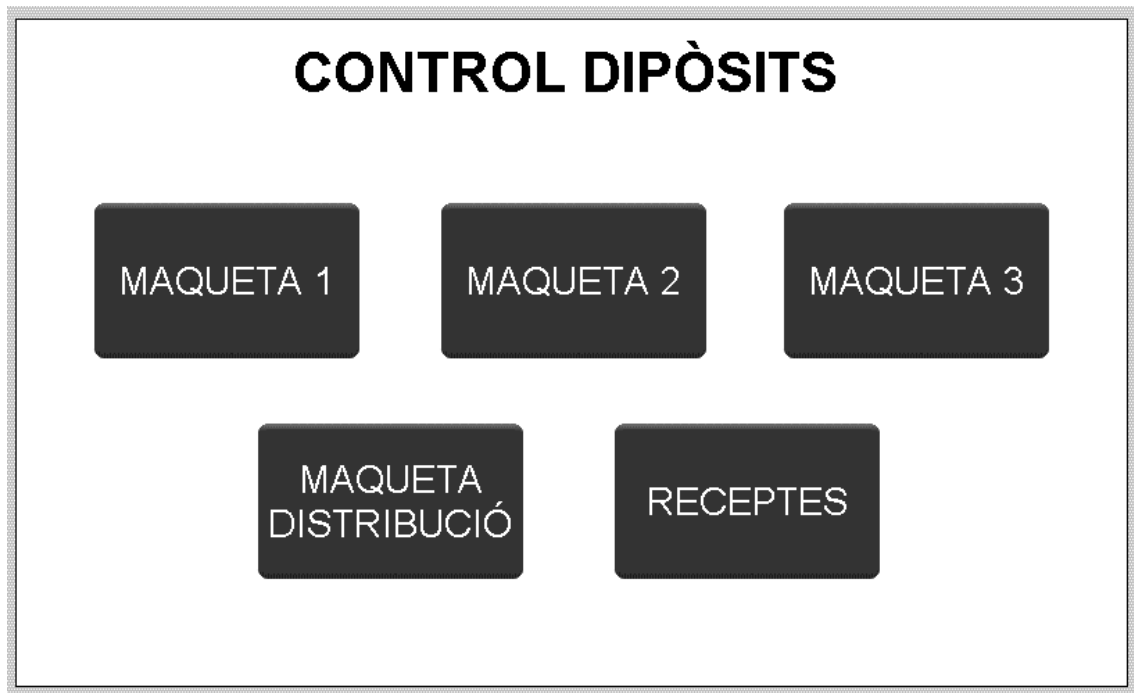


Imatge 67: Grafcet 70, recepta 6 maqueta distribució

## 4.3 Pantalles de visualització

### 4.3.1 Pantalla principal

Al iniciar la pantalla tàctil del projecte, se'ns mostra la següent pantalla:



*Imatge 68: Pantalla principal SCADA*

Com es pot observar a la Imatge 68 ens trobem 5 botons amb els que podem interactuar. Els tres botons superiors són els que ens permeten anar a les pantalles de cada una de les maquetes base, i els dos inferiors, el primer ens obra la pantalla de control de la maqueta de distribució i el segon ens permet accedir a la pantalla de les receptes per poder escollir la recepta que volem executar.

#### 4.3.2 Pantalla de receptes

Per escollir la recepta amb la que volem que treballi cada una de les maquetes, si polsem sobre del boto de “RECEPTES” se’ns obrirà la pantalla que es mostra seguidament:



Imatge 69: Pantalla receptes SCADA

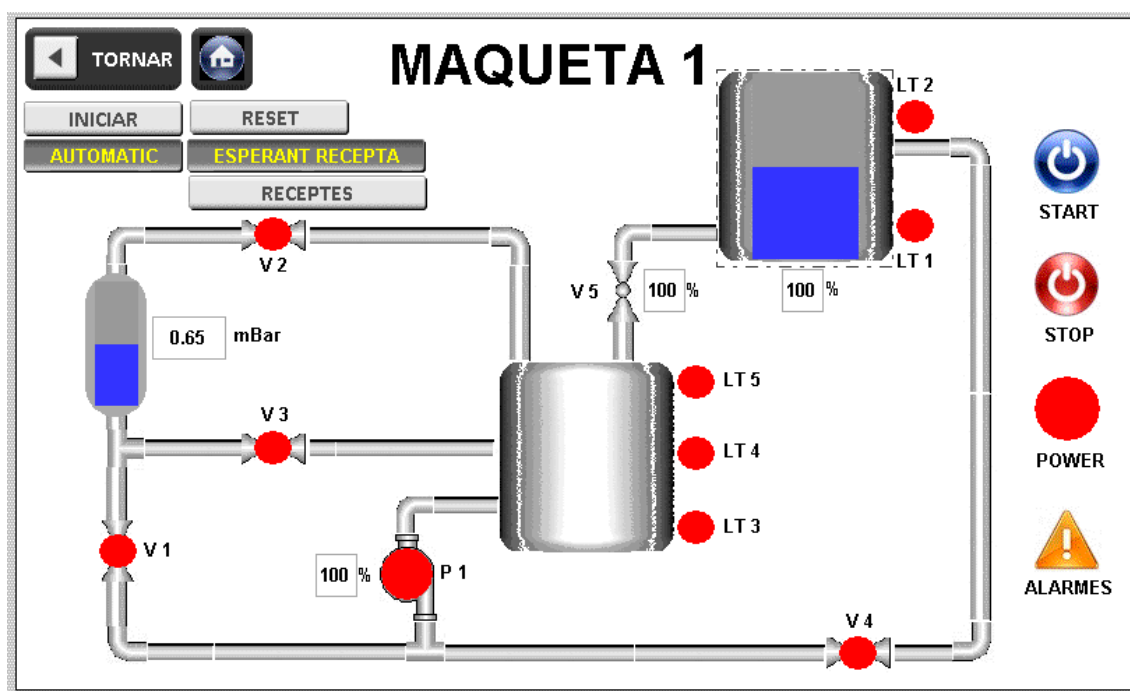
Aquesta pantalla està dividida en 3 parts. La part superior es on es troben els 6 botons per poder escollir entre cada una de les 6 receptes. Un cop escollida la recepta, a la part central es on es mostra una descripció simple de a quina maqueta va dirigida la recepta i quina es la seva funció. I a la part inferior de la pantalla es on es mostra la recepta que tenim escollida, en cas que no hi hagi cap recepta triada encara, ens apareixerà el missatge “ESPERANT RECEPTA”.

Finalment a la part superior esquerra de la pantalla es on es troben dos botons, el primer i el de mes a l’esquerra la funció que se li ha programat es la de retornar a la pantalla de on provenim, i el segon botó amb el símbol de la casa es per tornar directament a la pantalla principal sense haver d’anar tirant endarrere.

### 4.3.3 Pantalla de visualització maquetes 1, 2 i 3

Per controlar les maquetes base, es a dir les maquetes 1, 2 i 3, s'ha dissenyat una pantalla per cada maqueta, però que són iguals entre elles, amb la diferència que cada una controla la maqueta corresponent i per tant te les seves variables i senyals pròpies.

A continuació a la Imatge 70 es mostra la pantalla dissenyada per la maqueta 1 i s'expliquen les parts que la componen, només es descriu la de la maqueta en qüestió però es la mateixa, com eh dit anteriorment, per la maqueta 2 i la maqueta 3 canviant només les senyals i variables de control.



Imatge 70: Pantalla control i visualització maqueta 1

A la part superior esquerra de la pantalla es on trobem com en el cas anterior els dos botons per retrocedir i tornar a la pantalla des de on haguem accedit a aquesta pantalla, o directament si ho preferim anar directament a la pantalla principal.

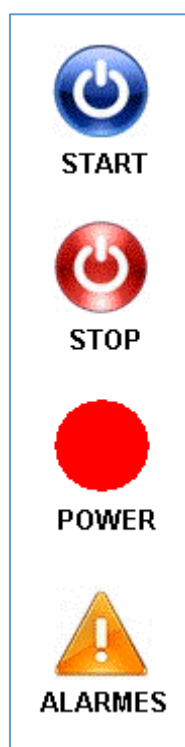
I per altre banda, es on hi ha el panell de control per iniciar o parar la recepta, poder fer un reset si hi ha alguna alarma, veure en quin estat es troba actualment la maqueta (manual o automàtic), poder visualitzar també la recepta que està carregada a la pròpia maqueta i per últim poder accedir a la pantalla de les receptes per canviar-la.



Imatge 71: Panell de control maqueta 1

L'altre panell de control, en aquest cas el principal, es troba a la part dreta de la pantalla. Aquí es on hi ha els pulsadors de marxa i parada per iniciar o parar el Grafcet principal de control, a més a més del LED de POWER que indica que la recepta s'està executant i el botó per poder accedir a la pantalla de les alarmes.

Els pulsadors de START i STOP tenen la mateixa funció que si activéssim o paréssim el programa des del teclat d'operacions físic de la pròpia maqueta. El LED en aquest cas també ens serveix per visualitzar el LED que hi ha físicament al teclat d'operacions però en aquest cas des de la pantalla tàctil, si la recepta no està en marxa el LED estarà de color vermell, en canvi si s'inicia el programa el LED passarà a estar de color verd.



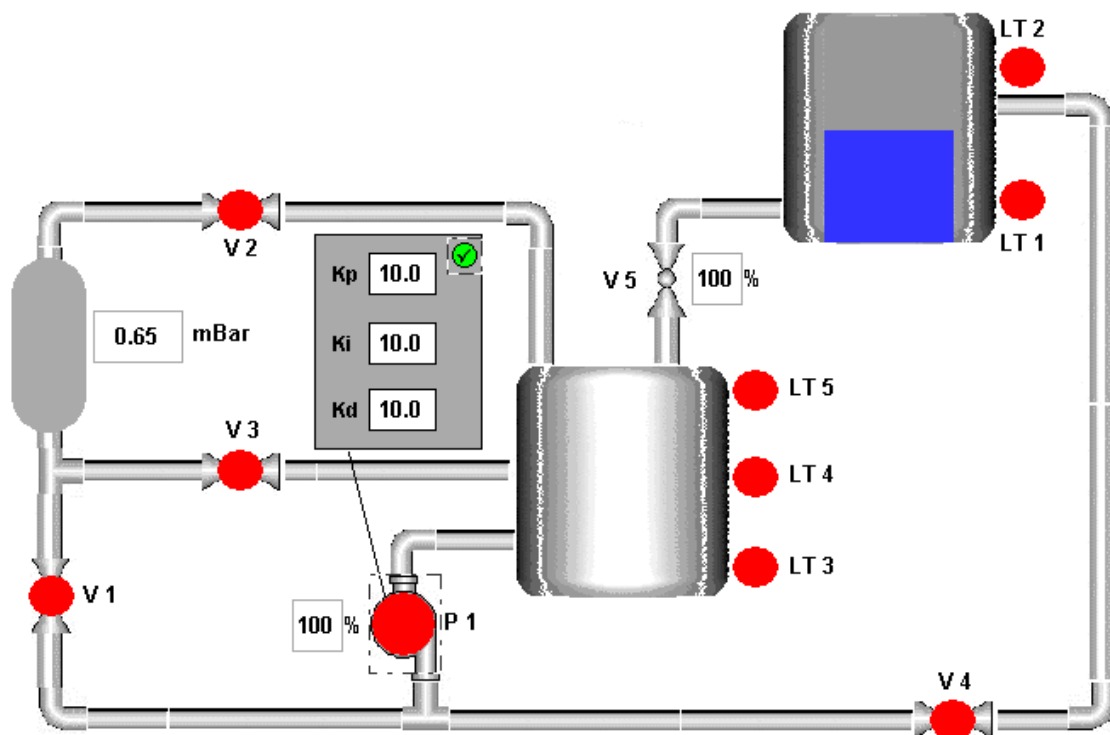
*Imatge 72: Panell de control principal maqueta 1*

I per últim, només queda la part central de la pantalla que és on es troba la visualització de la pròpia maqueta (veure Imatge 73).

Aquí es on hi ha representat l'esquema propi de la maqueta amb els dos dipòsits, les vàlvules, els sensors, la bomba i la finestra d'entrada i visualització de les constants del control PID de la bomba.

Igual que passa amb el LED de POWER, si les vàlvules no estan actuant o els sensors no estan detectant nivell, el color del cercle serà de color vermell, en canvi si alguna vàlvula està oberta o algun sensor està detectant, el color del mateix passarà a ser de color verd.

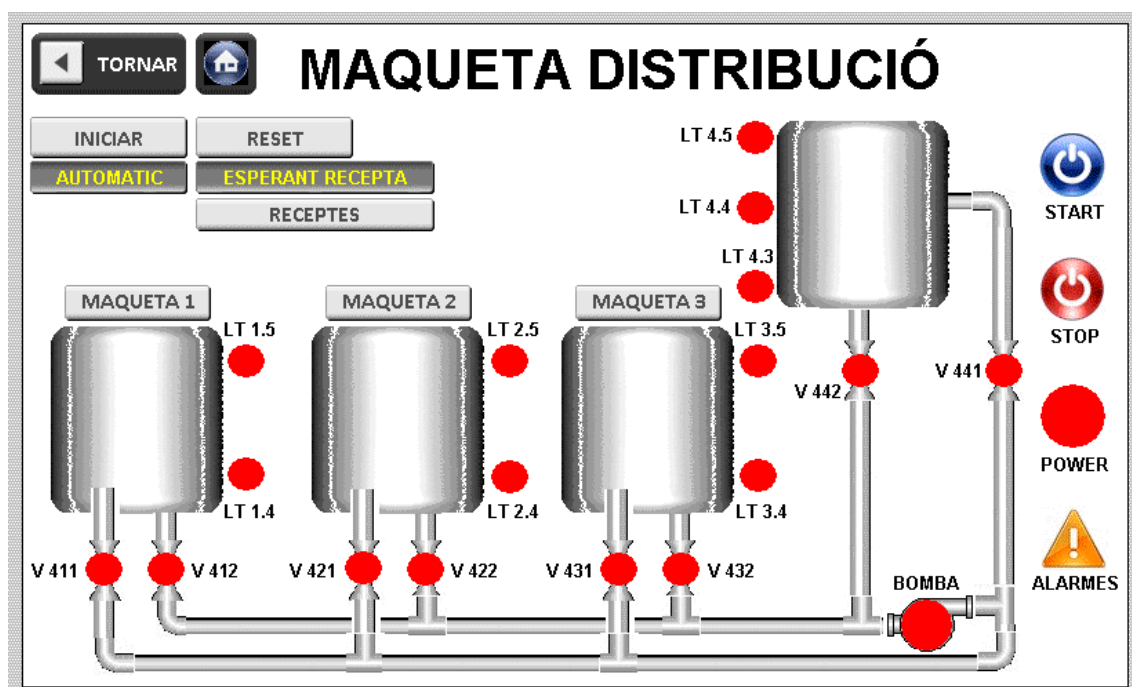




Imatge 73: Control i visualització esquematitzat maqueta 1

#### 4.3.4 Pantalla de visualització maqueta de distribució

A continuació es mostra la pantalla de visualització i control de la maqueta de distribució. Com es pot veure a la imatge següent hi ha dibuixats els 4 dipòsits, els 3 dipòsits inferiors de cada maqueta base i el dipòsit de la maqueta central.



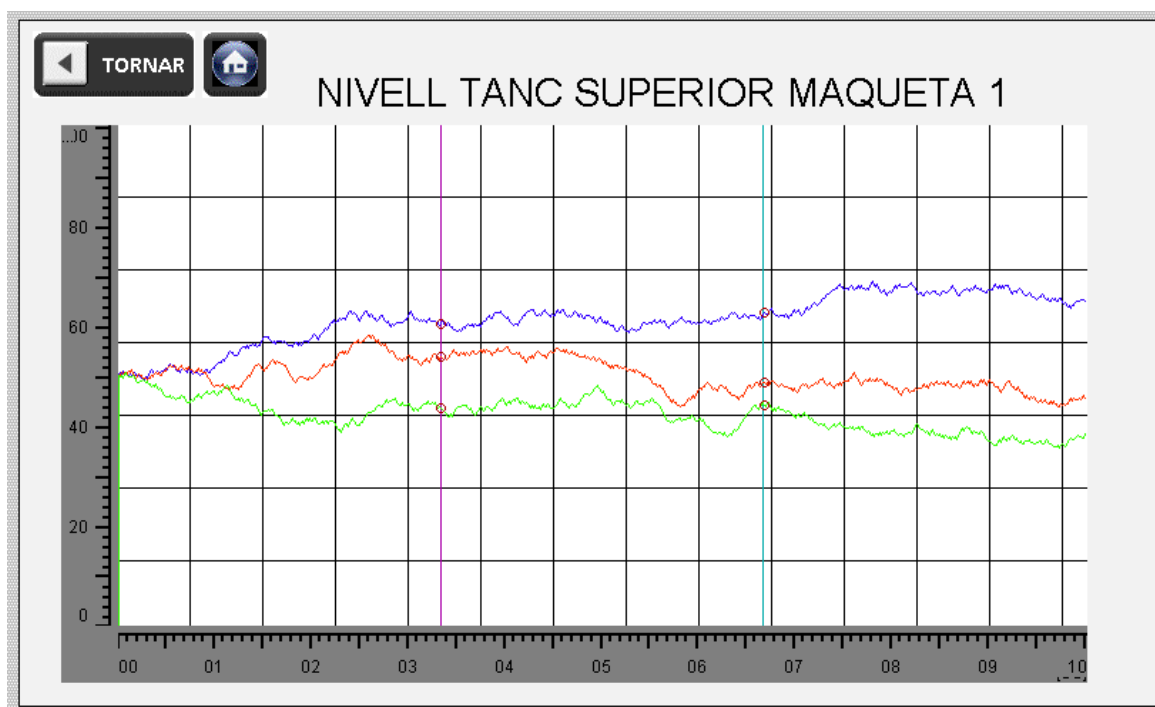
Imatge 74: Pantalla control i visualització maqueta distribució

Igual que en la pantalla de les maquetes base 1, 2 i 3 a la part superior esquerra es veu el panell de control per poder iniciar les receptes, eliminar les alarmes existents, veure el número de recepta o el mode actual en el que està treballant el programa (automàtic o manual).

La diferència es troba en la representació esquemàtica de la maqueta, on s'hi troben els 4 dipòsits amb les seves vàlvules corresponents, tant d'entrada com de sortida, la bomba i els sensors de nivell superior i inferior de cada tanca.

A més a més, des d'aquesta pantalla a part de poder retrocedir o anar a la pantalla principal com es pot fer en totes les altres pantalles, s'han habilitat uns pulsadors a sobre de cada dipòsit per poder accedir directament a la pantalla de la maqueta corresponent, sense haver de passar per la pantalla principal.

#### 4.3.5 Pantalla de gràfica de nivell i control PID



Imatge 75: Pantalla de visualització de la gràfica de nivell

La imatge anterior mostra el nivell, la consigna i el valor de sortida del controlador PID del dipòsit superior de les maquetes 1, 2 o 3 en funció de la maqueta en la que ens trobem.

Per accedir a aquesta pantalla s'hi accedeix fent clic sobre el dipòsit superior de cada maqueta. Un cop dintre tal i com hem dit es mostra la corba de nivell, de color blau, el valor de la consigna de color vermell i el valor actual de sortida del controlador PID de color verd.

Per sortir d'aquesta pantalla tenim dues formes, igual que passa amb les altres pantalles, o podem retrocedir fins a la pantalla de la que venim, fent clic a la fletxa o be podem anar directament a la pantalla principal fent clic a la icona de la casa.

## 5. POSSIBILITATS DE MILLORA

- **Control manual dels elements:**

Per poder controlar cada element per separat i poder fer proves o simplement omplir i buidar els dipòsits quan faci falta, i no haver d'obrir les vàlvules de bypass, seria interessant ampliar el diagrama de GEMMA i crear el GRAFCET per poder controlar via botons a la pantalla tàctil, cada element de les maquetes.

- **SCADA més visual:**

Es Podria realitzar un SCADA en el que les animacions de les vàlvules, els tubs, les bombes, etc. fossin més animades i vistoses.

- **Pantalla d'alarmes:**

La pantalla de les alarmes no ha donat temps de poder acabar-la. Seria interessant poder tenir una pantalla on visualitzar totes les alarmes actuals i poder reconèixer-les en cas que ja estiguin solucionades i tenir un històric de les alarmes passades.

- **Gràfics personalitzables:**

Un altre punt a tenir en compte seria, poder crear una pantalla on poder escollir d'una llista de variables, les que volem mostrar i poder-les veure totes juntes en un gràfic per poder comparar per exemple nivells dels diferents dipòsits o obertures de vàlvules i bombes.

- **Variabilitat de receptes:**

En aquest projecte s'ha dissenyat i programat la instal·lació perquè totes les maquetes puguin executar a la vegada la mateixa recepta, però no perquè puguin iniciar cadascuna una recepta diferent al mateix. Es podria modificar els GRAFCETS perquè cada maqueta pugui treballar independentment de les altres amb diferent numero de recepta i al mateix moment.

## 6. CONCLUSIONS

Al haver finalitzat aquest Projecte de Final de Grau puc estar satisfet d'haver complert amb els objectius inicials de manera satisfactòria.

Realitzant aquest projecte he pogut conèixer i comprendre el funcionament de diferents tecnologies Industrials, així com haver desenvolupat capacitats de síntesis, disseny, implementació, posta en marxa, programació de PLC, reparació d'averies, diagnòstic de fallades i elaboració de documentació tècnica.

He pogut conèixer el potencial del llenguatge GRAFCET, que és molt utilitzat per la programació de PLC's arran de la seva característica gràfica i fàcil disseny de processos Industrials, així com també eh pogut aprendre les grans capacitats i possibilitats de la programació amb llenguatge en text estructurat, igualment molt utilitzat avui en dia en la programació i creació de programes.

Per altre banda, m'agradaria destacar la gran dificultat que eh tingut a l'hora d'implementar la comunicació via OPC, ha estat un dels punts mes complicats de tot el projecte, ja que sense aquesta comunicació tot el projecte no podia avançar i no hagués aconseguit assolir part dels objectius marcats.

Finalment puc dir que he quedat satisfet amb la finalització d'aquest projecte, adquirint coneixements nous no vistos al llarg del grau cursat, i afermant alguns dels coneixements assimilats en algunes de les assignatures del Grau en Electrònica Industrial y Automàtica.

## 7. REFERÈNCIES





































- [1] B&R Industrial Automation GmbH, «Perfection in Automation | B&R Industrial Automation,» 2019. [En línia]. Available: <https://www.br-automation.com/es-es/>. [Últim accés: 5 març 2019].
- [2] B&R Industrial Automation GmbH, «6PPT30.0702-20W Basic Information,» 2019. [En línia]. Available: <https://www.br-automation.com/es-es/productos/visualizacion-y-gestion/power-panel-de-la-serie-tc/power-panel-de-la-serie-t/power-panel-t30/6ppt300702-20w/?noredirect=1>. [Últim accés: 5 març 2019].
- [3] PTC, Inc, «KEPServerEX Connectivity Platform | OPC Server | Kepware,» 2019. [En línia]. Available: <https://www.kepware.com/en-us/products/kepserverex/>. [Últim accés: 2019 març 2019].
- [4] Electro Industria, «SISTEMAS SCADA: La evolución de las plataformas de monitoreo y control,» març 2016. [En línia]. Available: <http://www.emb.cl/electroindustria/articulo.mvc?xid=2735>. [Últim accés: 26 març 2019].
- [5] ControlSIM, «SISTEMAS SCADA: SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS,» [En línia]. Available: <http://www.controlsim.cl/sistemas-scada/>. [Últim accés: 26 març 2019].
- [6] Scribbr, «Cómo hacer un TFG,» [En línia]. Available: <https://www.scribbr.es/category/estructura/>. [Últim accés: 26 març 2019].
- [7] J. M. Cano, «El autómatas programable, un universo de posibilidades en constante evolución,» Omron Electronics Iberia, 6 maig 2013. [En línia]. Available: <http://www.interempresas.net/Robotica/Articulos/108924-El-automata-programable-un-universo-de-posibilidades-en-constante-evolucion.html>. [Últim accés: 28 març 2019].
- [8] IDbox, «SCADA,» [En línia]. Available: <https://idboxrt.com/scada/>. [Últim accés: 27 març 2019].
- [9] C. F. Ramírez, «Sistemas SCADA, parte fundamental de la Industria 4.0,» 5 gener 2017. [En línia]. Available: <https://ciudadesdelfuturo.es/sistemas-scada-parte-fundamental-la-industria-4-0.php>. [Últim accés: 26 març 2019].
- [10] Kepware, «Connectivity Guide: Connecting to a B&R® Device,» febrer 2018. [En línia]. Available: <https://www.kepware.com/getattachment/b14d7b8a-ecd5-4f66-bcd1-b578f16a85bb/connecting-br-device-opc-ua.pdf>.

- 
- [11] keppure, «keppureverex Manual,» 2018. [En línia]. Available:  
<https://www.keppure.com/getattachment/5759d980-7641-42e8-b4fb-7293c835a2f9/keppureverex-manual.pdf>.
- [12] keppure, «KEppureServerEX® V6 Install Guide,» febrer 2019. [En línia]. Available:  
<https://www.keppure.com/getattachment/a2930ea4-a80c-4aaf-9a93-f3af86609fd1/install-guide.pdf>.
- [13] Unknown, «LINEA DE TIEMPO PLC,» 12 juny 2016. [En línia]. Available:  
<http://electronikwillsala.blogspot.com/2016/06/linea-de-tiempo-plc.html>. [Últim accés: 28 març 2019].
- [14] keppure, «OPC UA Client Driver,» 2018. [En línia]. Available:  
<https://www.keppure.com/getattachment/3c0687fe-fa1d-4833-8967-69cc63d291b8/opc-ua-client-manual.pdf>.

## 8. ANNEXES

### 8.1 Programes i variables maquetes base 1, 2 i 3

#### 8.1.1 Variables globals

Name	Type
 ALARM	BOOL
 A_M	BOOL
 Ci	BOOL
 ESTAT	UINT
 ETAPA_G100	UINT
 ETAPA_G120	UINT
 FT1	BOOL
 INICI	BOOL
 LT1	BOOL
 LT2	BOOL
 LT3	BOOL
 LT4	BOOL
 LT5	BOOL
 LT6	INT
 LT6_ACT	INT
 NOM_RECEPTA	STRING[80]
 OFF_ON	BOOL
 P1	INT
 PID_BOMBA_TD	REAL
 PID_BOMBA_TI	REAL
 PID_BOMBA_KP	REAL
 POWER	BOOL
 PT1	INT
 P_EMERGENCIA	BOOL
 RECEPTA	INT
 RESET	BOOL
 SPLT6	REAL
 START	BOOL
 START_PANTALLA	BOOL
 STOP	BOOL
 STOP_PANTALLA	BOOL
 V1	BOOL
 V2	BOOL
 V3	BOOL
 V4	BOOL
 V5	INT



### 8.1.2 Programa Graficets 100 i 120

Name	Type
ALM_EMERGENCIA	BOOL
TR0	BOOL
TR1	BOOL
TR2	BOOL
TR3	BOOL
TR4	BOOL
TR5	BOOL
TR6	BOOL
TR7	BOOL
TR8	BOOL
TR9	BOOL
TR10	BOOL
TR11	BOOL
TR12	BOOL
TR13	BOOL
R_TRIG_E100	R_TRIG
FLANC_E100	BOOL

```
PROGRAM _INIT
```

```
ETAPA_G100:= 100;
ETAPA_G120:= 120;
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
//DETECTAR FLANC DE PUJADA E100
R_TRIG_E100( CLK:= (ETAPA_G100 = 100) );
FLANC_E100 := R_TRIG_E100.Q;

// ACTUALITZACIÓ ENTRADES
TR0:= ((NOT A_M) AND ETAPA_G120 = 121 AND (START OR START_PANTALLA) AND Ci);
TR1:= ((NOT A_M) AND ETAPA_G120 = 121 AND (START OR START_PANTALLA) AND NOT
(Ci));
TR2:= (A_M AND ETAPA_G120 = 121 AND (START OR START_PANTALLA));
TR3:= Ci;
TR4:= (NOT(STOP) OR NOT(STOP_PANTALLA));
TR5:= ((NOT(STOP) OR NOT(STOP_PANTALLA)) OR ((NOT A_M)));
TR6:= (ESTAT = 0);
TR7:= Ci;
TR8:= OFF_ON;
TR9:= NOT(P_EMERGENCIA);
TR10:= P_EMERGENCIA;
TR11:= RESET;
TR12:= (ETAPA_G100 = 100);
TR13:= FLANC_E100;

// ETAPES DEL GRAFCET GEMMA
CASE ETAPA_G100 OF
100:
    IF (TR0 = TRUE) THEN
        ETAPA_G100:= 101;
    ELSIF (TR1 = TRUE) THEN
        ETAPA_G100:= 102;
    ELSIF (TR2 = TRUE) THEN
        ETAPA_G100:= 103;
    ELSE
        END_IF
```

```
101:      IF (TR4 = TRUE) THEN
            ETAPA_G100:= 104;
        END_IF

102:      IF (TR3 = TRUE) THEN
            ETAPA_G100:= 101;
        END_IF

103:      IF (TR5 = TRUE) THEN
            ETAPA_G100:= 105;
        END_IF

104:      IF (TR6 = TRUE) THEN
            ETAPA_G100:= 100;
        END_IF

105:      IF (TR7 = TRUE) THEN
            ETAPA_G100:= 100;
        END_IF

END_CASE

// ETAPES DEL GRAFCET DE LA PARADA DE SEURETAT
CASE ETAPA_G120 OF
120:      IF (TR8 = TRUE) THEN
            ETAPA_G120:= 121;
        END_IF

121:      IF (TR9 = TRUE) THEN
            ETAPA_G120:= 122;
        ELSIF (TR13 = TRUE) THEN
            ETAPA_G120:= 120;
        ELSE
        END_IF

122:      IF (TR10 = TRUE) THEN
            ETAPA_G120:= 123;
        END_IF

123:      IF (TR11 = TRUE) THEN
            ETAPA_G120:= 124;
        END_IF

124:      IF (TR12 = TRUE) THEN
            ETAPA_G120:= 120;
        END_IF

END_CASE

// SORTIDES
IF (ETAPA_G100 = 102) OR (ETAPA_G100 = 105) THEN
    IF (LT1 = TRUE) THEN
        Ci:= FALSE;
        V5:= 32767;
    ELSE
        Ci:= TRUE;
        V5:= 0;
    END_IF
END_IF

IF (ETAPA_G120 = 122) THEN
    ETAPA_G100:= 0;
    ESTAT:= 0;
    ALARM:= TRUE;
    ALM_EMERGENCIA:= TRUE;
END_IF
```

```
IF (ETAPA_G120 = 124) THEN
    ETAPA_G100:= 105;
    ESTAT:= 0;
    ALARM:= FALSE;
    ALM_EMERGENCIA:= FALSE;
END_IF

END_PROGRAM

PROGRAM _EXIT

END_PROGRAM
```

### 8.1.3 Programa Espera de recepta

Name	Type
◆ INTERMITENCIA	TON
◆ T0	BOOL
◆ T1	BOOL
◆ T2	BOOL
◆ T3	BOOL
◆ T4	BOOL
◆ T5	BOOL
◆ T6	BOOL
◆ TIMER_1	TON
◆ TIMER_2	TON
◆ TIMER_3	TON

```
PROGRAM _INIT

    ESTAT:= 0;

END_PROGRAM

PROGRAM _CYCLIC

    // ASSIGNACIÓ DE TRANSICIONS
    T0:= INICI AND (ETAPA_G100 = 101);
    T4:= TIMER_1.Q;
    T5:= TIMER_2.Q;
    T6:= TIMER_3.Q;

    IF (RECEPTA = 1) THEN
        T1:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 1
        T2:= FALSE;
        T3:= FALSE;
    ELSIF (RECEPTA = 2) THEN
        T2:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 2
        T1:= FALSE;
        T3:= FALSE;
    ELSIF (RECEPTA = 3) THEN
        T3:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 3
        T1:= FALSE;
        T2:= FALSE;
    ELSE
        T1:= FALSE;
        T2:= FALSE;
        T3:= FALSE;
    END_IF
```

```
// EVOLUCIÓ D'ESTATS
CASE ESTAT OF
  0:
    IF (T0 = TRUE) THEN
      ESTAT:= 1;
    END_IF
  1:
    IF (T1 = TRUE) THEN
      ESTAT:= 2;
    ELSIF (T2 = TRUE) THEN
      ESTAT:= 3;
    ELSIF (T3 = TRUE) THEN
      ESTAT:= 4;
    END_IF
  2:
    IF (T4 = TRUE) THEN
      ESTAT:= 10;
    END_IF
  3:
    IF (T5 = TRUE) THEN
      ESTAT:= 20;
    END_IF
  4:
    IF (T6 = TRUE) THEN
      ESTAT:= 30;
    END_IF
END_CASE

// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 2), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 3), PT:= T#1s);
TIMER_3(IN:= (ESTAT = 4), PT:= T#1s);

// SORTIDES
IF (ESTAT = 0) THEN
  POWER:= FALSE;
  P1:= 0;
  V1:= FALSE;
  V2:= FALSE;
  V3:= FALSE;
  V4:= FALSE;
  V5:= 0;
END_IF

IF (ESTAT = 1) THEN
  NOM_RECEPTA:= 'ESPERANT RECEPTA';
  POWER:= FALSE;
END_IF

IF (ESTAT = 2) THEN
  NOM_RECEPTA:= 'RECEPTA 1';
  POWER:= FALSE;
END_IF

IF (ESTAT = 3) THEN
  NOM_RECEPTA:= 'RECEPTA 2';
  POWER:= FALSE;
END_IF








IF (ESTAT = 4) THEN
  NOM_RECEPTA:= 'RECEPTA 3';
  POWER:= FALSE;
END_IF

END_PROGRAM

PROGRAM _EXIT

END_PROGRAM
```

#### 8.1.4 Programa Recepta 1

Name	Type
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 TIMER_1	TON
 TIMER_2	TON
 V5_value	INT

```
PROGRAM _INIT
```

```
V1:= FALSE;  
V2:= FALSE;  
V3:= FALSE;  
V4:= FALSE;  
V5:= 0;  
P1:= 0;
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
// ACTUALITZACIÓ ENTRADES  
T0:= TIMER_1.Q AND (NOT LT1);  
T1:= LT1 AND (NOT LT4);  
T2:= LT4 AND (NOT LT1);  
T3:= TIMER_2.Q;
```

```
// ETAPES DEL GRAFCET  
CASE ESTAT OF  
  10:  
    IF (T0 = TRUE) THEN  
      ESTAT:= 11;  
    END_IF  
  11:  
    IF (T1 = TRUE) THEN  
      ESTAT:= 12;  
    END_IF  
  12:  
    IF (T2 = TRUE) THEN  
      ESTAT:= 13;  
    END_IF  
  13:  
    IF (T3 = TRUE) THEN  
      ESTAT:= 0;  
    END_IF  
END_CASE
```

```
// TEMPORITZADORS  
TIMER_1(IN:= (ESTAT = 10), PT:= T#1s);  
TIMER_2(IN:= (ESTAT = 13), PT:= T#1s);
```

```
// SORTIDES  
IF (ESTAT = 10) THEN  
  POWER:= FALSE;  
END_IF
```

```
IF (ESTAT = 11) THEN  
  P1:= 32767;  
  V4:= TRUE;  
  POWER:= TRUE;  
END_IF
```

```
IF (ESTAT = 12) THEN
```

```

P1:= 0;
V5_value:= 32767;
V4:= FALSE;
V5:= V5_value;
END_IF

IF (ESTAT = 13) THEN
    V5_value:= 0;
    V5:= V5_value;
END_IF
























END_PROGRAM

PROGRAM _EXIT

END_PROGRAM

```

### 8.1.5 Programa Recepta 2

Name	Type
 actValue	INT
 baseLCCounter	LCCounter
 LCPIDpara_0	LCPIDpara
 LCPID_0	LCPID
 LCRPT2_0	LCRPT2
 manipulatedVar	INT
 paraEnter	BOOL
 setValue	INT
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 TIMER_1	TON
 TIMER_2	TON
 V5_value	INT
 LT6_50	INT
 start_PID	BOOL
 T7	BOOL
 T8	BOOL
 T9	BOOL

```

PROGRAM _INIT

(* Init variables *)
LT6_50:= 5000; (* Nivell al 30% *)
V1:= FALSE;
V2:= FALSE;
V3:= FALSE;
V4:= FALSE;
V5:= 0;
P1:= 0;
setValue := 0;

```

```
actValue := 0;
PID_BOMBA_KP:= 0.5;
PID_BOMBA_TI:= 150.0;
PID_BOMBA_TD:= 0.0;
start_PID := FALSE;
paraEnter := FALSE;

(* PID parameters *)
LCPIDpara_0.Y_max      := 32767;
LCPIDpara_0.Y_min      := 0;
LCPIDpara_0.dY_max     := 0.0;
LCPIDpara_0.Kp         := PID_BOMBA_KP;
LCPIDpara_0.Tn         := PID_BOMBA_TI;
LCPIDpara_0.Tv         := PID_BOMBA_TD;
LCPIDpara_0.Tf         := 0.0;
LCPIDpara_0.Kw         := 1.0;
LCPIDpara_0.Kfbk       := 1.0; (* windup damping enabled *)
LCPIDpara_0.fbk_mode   := LCPID_FBK_MODE_INTERN;
LCPIDpara_0.d_mode     := LCPID_D_MODE_X;
LCPIDpara_0.calc_mode  := LCPID_CALC_MODE_FAST;

(* Parameters for PID controller *)
LCPID_0.A              := 0;
LCPID_0.Y_man          := 0;
LCPID_0.Y_fbk         := 0;
LCPID_0.hold_I         := FALSE;
LCPID_0.out_mode       := LCPID_OUT_MODE_AUTO;

(* Parameters for PT2 element *)
LCRPT2_0.V             := 0.039;
LCRPT2_0.T1            := 0.9;
LCRPT2_0.T2            := 15.3;
LCRPT2_0.y_set         := 0.0;
LCRPT2_0.set           := 0;

END_PROGRAM

PROGRAM _CYCLIC

LT6_ACT:= 32767 - LT6;
actValue := LT6_ACT;

LCPIDpara_0.Kp := PID_BOMBA_KP;
LCPIDpara_0.Tn := PID_BOMBA_TI;
LCPIDpara_0.Tv := PID_BOMBA_TD;
paraEnter:= TRUE;

(* Time base counter *)
baseLCCounter();
(* PID parameters *)
LCPIDpara_0.enable := start_PID;
LCPIDpara_0.enter  := paraEnter;
LCPIDpara_0(); (* LCPIDpara function block call *)
paraEnter := 0;
(* PID controller *)
LCPID_0.enable := start_PID;
LCPID_0.ident  := LCPIDpara_0.ident; (* ident of PIDpara -> provides parameters
(Kp, Tn, Tv, ...) *)
LCPID_0.W      := setValue;
LCPID_0.X      := actValue;
LCPID_0.basetime := baseLCCounter.msct;
LCPID_0(); (* LCPID function block call *)
manipulatedVar := LCPID_0.Y;
//(* PT2 element *)
// LCRPT2_0.enable := start_PID;
// LCRPT2_0.x      := manipulatedVar;
// LCRPT2_0(); (* LCRPT2 function block call *)

paraEnter:= FALSE;
```

```
// ACTUALITZACIÓ ENTRADES

T0:= (TIMER_1.Q) AND (NOT LT1);
T1:= LT1;
T2:= LT6 >= LT6_50;
T3:= (NOT LT3) OR TIMER_2.Q;
T4:= LT3 AND (NOT LT1);

// ETAPES DEL GRAFCET
CASE ESTAT OF
  20:
    IF (T0 = TRUE) THEN
      ESTAT:= 21;
    END_IF
  21:
    IF (T1 = TRUE) THEN
      ESTAT:= 22;
    END_IF
  22:
    IF (T2 = TRUE) THEN
      ESTAT:= 23;
    END_IF
  23:
    IF (T3 = TRUE) THEN
      ESTAT:= 24;
    END_IF
  24:
    IF (T4= TRUE) THEN
      ESTAT:= 0;
    END_IF
END_CASE

// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 20), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 23), PT:= T#300s);

// SORTIDES
IF (ESTAT = 20) THEN
  POWER:= FALSE;
END_IF

IF (ESTAT = 21) THEN
  P1:= 32767;
  V4:= TRUE;
  POWER:= TRUE;
END_IF

IF (ESTAT = 22) THEN
  P1:= 32767;
  setValue:= LT6_50;
END_IF











IF (ESTAT = 23) THEN
  start_PID:= TRUE;
  paraEnter:= TRUE;
  V5_value:= 32767;
  V5:= V5_value;
  P1:= manipulatedVar;
END_IF

IF (ESTAT = 24) THEN
  start_PID:= FALSE;
  P1:= 0;
  V5_value:= 32767;
  V5:= V5_value;
  V4:= FALSE;
END_IF

END_PROGRAM
```



### 8.1.6 Programa Recepta 3

Name	Type
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 TIMER_1	TON
 TIMER_2	TON
 TIMER_3	TON
 TIMER_5	TON
 TIMER_4	TON

```
PROGRAM _INIT
```

```
V1:= FALSE;
V2:= FALSE;
V3:= FALSE;
V4:= FALSE;
V5:= 0;
P1:= 0;
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
(* ACTUALITZACIÓ ENTRADES *)

T0:= TIMER_1.Q AND (ETAPA_G100 = 101);
T1:= TIMER_2.Q;
T2:= (PT1 >= 32760) AND TIMER_3.Q;
T3:= TIMER_4.Q;
T4:= TIMER_5.Q;

(*ETAPES DEL GRAFCET*)

CASE ESTAT OF
  30:
    IF (T0 = TRUE) THEN
      ESTAT:= 31;
    END_IF
  31:
    IF (T1 = TRUE) THEN
      ESTAT:= 32;
    END_IF
  32:
    IF (T2 = TRUE) THEN
      ESTAT:= 33;
    END_IF
  33:
    IF (T3 = TRUE) THEN
      ESTAT:= 34;
    END_IF
  34:
    IF (T4 = TRUE) THEN
      ESTAT:= 0;
    END_IF
END_CASE
```

```
// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 30), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 31), PT:= T#20s);
TIMER_3(IN:= (ESTAT = 32), PT:= T#10s);
TIMER_4(IN:= (ESTAT = 33), PT:= T#10s);
TIMER_5(IN:= (ESTAT = 34), PT:= T#5s);

(*SORTIDES*)
IF (ESTAT = 30) THEN
    POWER:= FALSE;
    V3:= FALSE;
END_IF

IF (ESTAT = 31) THEN
    V3:= TRUE;
    V2:= TRUE;
    POWER:= TRUE;
END_IF

IF (ESTAT = 32) THEN
    P1:= 32767;
    V1:= TRUE;
    V3:= FALSE;
    V2:= FALSE;
END_IF

IF (ESTAT = 33) THEN
    P1:= 0;
    V1:= FALSE;
    V2:= TRUE;
    V3:= TRUE;
END_IF

IF (ESTAT = 34) THEN
    V3:= FALSE;
    V2:= FALSE;
END_IF

END_PROGRAM
```

## 8.2 Programes i variables maqueta distribució

### 8.2.1 Variables globals

Name	Type
ALARM	BOOL
ALARMA_1	BOOL
ALARMA_2	BOOL
ALARMA_3	BOOL
ALARMA_ACTIVA_1	UINT
ALARMA_ACTIVA_2	UINT
ALARMA_ACTIVA_3	UINT
ALARM_EMERGENCIA_1	BOOL
ALARM_PRESSIO_1	BOOL
AUTO_MAN	BOOL
A_M_1_ESTAT	BOOL
A_M_2_ESTAT	BOOL
A_M_3_ESTAT	BOOL
Ci	BOOL
ESTAT	UINT
ETAPA_G200	UINT
ETAPA_G220	UINT
I4	BOOL
I5	BOOL
LT43	BOOL
I7	BOOL
INICIAR	BOOL
INICIAR_1	BOOL
INICIAR_2	BOOL
INICIAR_3	BOOL
LT14	BOOL
LT15	BOOL
LT24	BOOL
LT25	BOOL
LT34	BOOL
LT35	BOOL
LT44	BOOL
LT45	BOOL
LT_11	BOOL
LT_12	BOOL
LT_13	BOOL
LT_14	BOOL
LT_15	BOOL
LT_16	INT

◆ LT_21	BOOL
◆ LT_22	BOOL
◆ LT_23	BOOL
◆ LT_24	BOOL
◆ LT_25	BOOL
◆ LT_26	INT
◆ LT_31	BOOL
◆ LT_32	BOOL
◆ LT_33	BOOL
◆ LT_34	BOOL
◆ LT_35	BOOL
◆ LT_36	INT
◆ L_RESET	BOOL
◆ L_START	BOOL
◆ POWER_1	BOOL
◆ POWER_2	BOOL
◆ POWER_3	BOOL
◆ PT_11	INT
◆ PT_21	INT
◆ PID_OUT_2	INT
◆ PID_OUT_3	INT
◆ PID_OUT_1	INT
◆ CONSIGNA_2	INT
◆ CONSIGNA_3	INT
◆ CONSIGNA_1	INT
◆ PT_31	INT
◆ PUMP	BOOL
◆ P_11	INT
◆ P_21	INT
◆ P_31	INT
◆ P_EMERGENCIA	BOOL
◆ Q1	BOOL
◆ Q2	BOOL
◆ Q4	BOOL
◆ Q5	BOOL
◆ Q6	BOOL
◆ RECEPTE	INT
◆ RESET	BOOL
◆ RESET_1	BOOL
◆ RESET_2	BOOL
◆ RESET_3	BOOL
◆ RESET_PANTALLA	BOOL
◆ SPLT61	REAL

◆ START	BOOL
◆ START_1	BOOL
◆ START_2	BOOL
◆ START_3	BOOL
◆ START_PANTALLA	BOOL
◆ STOP	BOOL
◆ STOP_1	BOOL
◆ STOP_2	BOOL
◆ STOP_3	BOOL
◆ STOP_PANTALLA	BOOL
◆ V411	BOOL
◆ V412	BOOL
◆ V421	BOOL
◆ V422	BOOL
◆ V431	BOOL
◆ V432	BOOL
◆ V441	BOOL
◆ V442	BOOL
◆ V_11	BOOL
◆ V_12	BOOL
◆ V_13	BOOL
◆ V_14	BOOL
◆ V_15	UINT
◆ V_21	BOOL
◆ V_22	BOOL
◆ V_23	BOOL
◆ V_24	BOOL
◆ V_25	UINT
◆ V_31	BOOL
◆ V_32	BOOL
◆ V_33	BOOL
◆ V_34	BOOL
◆ V_35	UINT

## 8.2.2 Programa Graficets 200 i 220

Name	Type
ALM_EMERGENCIA	BOOL
TR0	BOOL
TR1	BOOL
TR2	BOOL
TR3	BOOL
TR4	BOOL
TR5	BOOL
TR6	BOOL
TR7	BOOL
TR8	BOOL
TR9	BOOL
TR10	BOOL
TR11	BOOL
TR12	BOOL
TR13	BOOL
R_TRIG_E200	R_TRIG
FLANC_E200	BOOL

PROGRAM \_INIT

```
ETAPA_G200:= 200;
ETAPA_G220:= 220;
```

END\_PROGRAM

PROGRAM \_CYCLIC

```
//DETECTAR FLANC DE PUJADA E200
R_TRIG_E200( CLK:= ETAPA_G200 = 200 );
FLANC_E200 := R_TRIG_E200.Q;

// ACTUALITZACIÓ ENTRADES
TR0:= ((NOT AUTO_MAN) AND (ETAPA_G220 = 221) AND (START OR START_PANTALLA) AND
Ci);
TR1:= ((NOT AUTO_MAN) AND (ETAPA_G220 = 221) AND (START OR START_PANTALLA) AND
NOT (Ci));
TR2:= (AUTO_MAN AND (ETAPA_G220 = 221) AND (START OR START_PANTALLA));
TR3:= Ci;
TR4:= (NOT(STOP) AND NOT(STOP_PANTALLA));
TR5:= ((NOT(STOP) OR NOT(STOP_PANTALLA)) OR ((NOT AUTO_MAN)));
TR6:= (ESTAT = 40);
TR7:= Ci;
TR8:= (START OR START_PANTALLA);
TR9:= NOT(P_EMERGENCIA);
TR10:= P_EMERGENCIA;
TR11:= RESET OR RESET_PANTALLA;
TR12:= (ETAPA_G200 = 200);
TR13:= FLANC_E200;
```

```
// ETAPES DEL GRAFCET GEMMA
CASE ETAPA_G200 OF
  200:
    IF (TR0 = TRUE) THEN
      ETAPA_G200:= 201;
    ELSIF (TR1 = TRUE) THEN
      ETAPA_G200:= 202;
    ELSIF (TR2 = TRUE) THEN
      ETAPA_G200:= 203;
    ELSE
      END_IF
  201:
    IF (TR4 = TRUE) THEN
      ETAPA_G200:= 204;
    END_IF
  202:
    IF (TR3 = TRUE) THEN
      ETAPA_G200:= 201;
    END_IF
  203:
    IF (TR5 = TRUE) THEN
      ETAPA_G200:= 205;
    END_IF
  204:
    IF (TR6 = TRUE) THEN
      ETAPA_G200:= 200;
    END_IF
  205:
    IF (TR7 = TRUE) THEN
      ETAPA_G200:= 200;
    END_IF
END_CASE

// ETAPES DEL GRAFCET DE LA PARADA DE SEURETAT
CASE ETAPA_G220 OF
  220:
    IF (TR8 = TRUE) THEN
      ETAPA_G220:= 221;
    END_IF
  221:
    IF (TR9 = TRUE) THEN
      ETAPA_G220:= 222;
    ELSIF (TR13 = TRUE) THEN
      ETAPA_G220:= 220;
    ELSE
      END_IF
  222:
    IF (TR10 = TRUE) THEN
      ETAPA_G220:= 223;
    END_IF
  223:
    IF (TR11 = TRUE) THEN
      ETAPA_G220:= 224;
    END_IF
  224:
    IF (TR12 = TRUE) THEN
      ETAPA_G220:= 220;
    END_IF
END_CASE

// SORTIDES
IF (ETAPA_G200 = 202) OR (ETAPA_G200 = 205) THEN
//CONDICIONS INICIALS
IF ((LT14 = TRUE) OR (LT24 = TRUE) OR (LT34 = TRUE) OR (LT43 = TRUE)) THEN
  Ci:= TRUE;
ELSE
  Ci:= FALSE;
END_IF
END_IF
```

```
IF (ETAPA_G220 = 222) THEN
    ETAPA_G200:= 200;
    ESTAT := 40;
    ALARM:= TRUE;
    ALM_EMERGENCIA:= TRUE;
END_IF

IF (ETAPA_G220 = 224) THEN
    ETAPA_G200:= 205;
    ESTAT:= 40;
    ALARM:= FALSE;
    ALM_EMERGENCIA:= FALSE;
END_IF

END_PROGRAM
```

### 8.2.3 Programa Espera de recepta

Name	Type
◆ INTERMITENCIA	TON
◆ T0	BOOL
◆ T1	BOOL
◆ T2	BOOL
◆ T3	BOOL
◆ T4	BOOL
◆ T5	BOOL
◆ T6	BOOL
◆ TIMER_1	TON
◆ TIMER_2	TON
◆ TIMER_3	TON

```
PROGRAM _INIT

    ESTAT:= 40;

END_PROGRAM

PROGRAM _CYCLIC

    // ASSIGNACIÓ DE TRANSICIONS
    T0:= INICIAR AND ETAPA_G200 = 201;
    T4:= TIMER_1.Q;
    T5:= TIMER_2.Q;
    T6:= TIMER_3.Q;

    IF (RECEPTA = 4) THEN
        T1:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 4
        T2:= FALSE;
        T3:= FALSE;
    ELSIF (RECEPTA = 5) THEN
        T2:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 5
        T1:= FALSE;
        T3:= FALSE;
    ELSIF (RECEPTA = 6) THEN
        T3:= TRUE;      //TRANSICIÓ PER ACTIVAR LA RECEPTA 6
        T1:= FALSE;
        T2:= FALSE;
    ELSE


```



```
T1:= FALSE;
T2:= FALSE;
T3:= FALSE;

END_IF

// EVOLUCIÓ D'ESTATS
CASE ESTAT OF
40:
    IF (T0 = TRUE) THEN
        ESTAT:= 41;
    END_IF
41:
    IF (T1 = TRUE) THEN
        ESTAT:= 42;
    ELSIF (T2 = TRUE) THEN
        ESTAT:= 43;
    ELSIF (T3 = TRUE) THEN
        ESTAT:= 44;
    END_IF
42:
    IF (T4 = TRUE) THEN
        ESTAT:= 50;
    END_IF
43:
    IF (T5 = TRUE) THEN
        ESTAT:= 60;
    END_IF
44:
    IF (T6 = TRUE) THEN
        ESTAT:= 70;
    END_IF
END_CASE

// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 42), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 43), PT:= T#1s);
TIMER_3(IN:= (ESTAT = 44), PT:= T#1s);
INTERMITENCIA(IN:= (ESTAT = 41) AND NOT (INTERMITENCIA.Q) ,PT:= T#0.5s);

// SORTIDES
IF (ESTAT = 40) THEN
    L_START:= FALSE;
END_IF

IF (ESTAT = 41) THEN
    IF INTERMITENCIA.Q = TRUE THEN
        L_START:= TRUE;
    ELSE
        L_START:= FALSE;
    END_IF
END_IF

IF (ESTAT = 42) THEN
    L_START:= FALSE;
END_IF

IF (ESTAT = 43) THEN
    L_START:= FALSE;
END_IF

IF (ESTAT = 44) THEN
    L_START:= FALSE;
END_IF

END_PROGRAM
```

## 8.2.4 Programa de Receptes

Name	Type
◆ NUM_RECEPTA	UINT
◆ NOM_RECEPTA	STRING[80]

```
PROGRAM _INIT
```

```
    RECEPTA:= 0;  
    NUM_RECEPTA := 0;  
    NOM_RECEPTA := '';
```

```
END_PROGRAM
```












```
PROGRAM _CYCLIC
```

```
    CASE NUM_RECEPTA OF  
        1:  
            RECEPTA:= 1;  
            NOM_RECEPTA:= 'RECEPTA 1';  
        2:  
            RECEPTA:= 2;  
            NOM_RECEPTA:= 'RECEPTA 2';  
        3:  
            RECEPTA:= 3;  
            NOM_RECEPTA:= 'RECEPTA 3';  
        4:  
            RECEPTA:= 4;  
            NOM_RECEPTA:= 'RECEPTA 4';  
        5:  
            RECEPTA:= 5;  
            NOM_RECEPTA:= 'RECEPTA 5';  
        6:  
            RECEPTA:= 6;  
            NOM_RECEPTA:= 'RECEPTA 6';
```

```
    END_CASE
```

```
END_PROGRAM
```

### 8.2.5 Programa Recepta 4

Name	Type
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 T7	BOOL
 TIMER_1	TON
 TIMER_2	TON
 TIMER_3	TON

```
PROGRAM _INIT
```

```
V411:= FALSE;  
V412:= FALSE;  
V441:= FALSE;  
V442:= FALSE;  
PUMP:= FALSE;  
L_START:= FALSE;
```

```
END_PROGRAM
```

```
PROGRAM _CYCLIC
```

```
// ACTUALITZACIÓ ENTRADES  
T0:= TIMER_1.Q AND (NOT LT14) AND LT44;  
T1:= NOT LT44 OR NOT LT43;  
T2:= NOT LT14;  
T3:= TIMER_2.Q;  
T4:= TIMER_1.Q AND (NOT LT44) AND LT14;  
T5:= NOT LT14;  
T6:= NOT LT44 OR NOT LT43;  
T7:= TIMER_3.Q;
```

```
// ETAPES DEL GRAFCET
```

```
CASE ESTAT OF  
  50:  
    IF (T0 = TRUE) THEN  
      ESTAT:= 51;  
    ELSIF (T4 = TRUE) THEN  
      ESTAT:= 54;  
    END_IF  
  51:  
    IF (T1 = TRUE) THEN  
      ESTAT:= 52;  
    END_IF  
  52:  
    IF (T2 = TRUE) THEN  
      ESTAT:= 53;  
    END_IF  
  53:  
    IF (T3 = TRUE) THEN  
      ESTAT:= 40;  
    END_IF
```

```
54:      IF (T5 = TRUE) THEN
          ESTAT:= 55;
      END_IF
55:      IF (T6 = TRUE) THEN
          ESTAT:= 56;
      END_IF
56:      IF (T7 = TRUE) THEN
          ESTAT:= 40;
      END_IF
END_CASE

// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 50), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 53), PT:= T#1s);
TIMER_3(IN:= (ESTAT = 56), PT:= T#1s);

// SORTIDES
IF (ESTAT = 50) THEN
    L_START:= FALSE;
END_IF

IF (ESTAT = 51) THEN
    V411:= TRUE;
    V442:= TRUE;
    PUMP:= TRUE;
    L_START:= TRUE;
END_IF

IF (ESTAT = 52) THEN
    V412:= TRUE;
    V441:= TRUE;
    V442:= FALSE;
    V411:= FALSE;
END_IF

IF (ESTAT = 53) THEN
    PUMP:= FALSE;
    V412:= FALSE;
    V441:= FALSE;
END_IF












IF (ESTAT = 54) THEN
    V412:= TRUE;
    V441:= TRUE;
    PUMP:= TRUE;
    L_START:= TRUE;
END_IF

IF (ESTAT = 55) THEN
    V412:= FALSE;
    V441:= FALSE;
    V442:= TRUE;
    V411:= TRUE;
END_IF

IF (ESTAT = 56) THEN
    PUMP:= FALSE;
    V411:= FALSE;
    V442:= FALSE;
END_IF

END_PROGRAM
```

## 8.2.6 Programa Recepta 5

Name	Type
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 T7	BOOL
 TIMER_1	TON
 TIMER_2	TON
 TIMER_3	TON

PROGRAM \_INIT

```
V421:= FALSE;
V422:= FALSE;
V441:= FALSE;
V442:= FALSE;
PUMP:= FALSE;
L_START:= FALSE;
```

END\_PROGRAM

PROGRAM \_CYCLIC

```
// ACTUALITZACIÓ ENTRADES
T0:= TIMER_1.Q AND (NOT LT24) AND LT44;
T1:= NOT LT44 OR NOT LT43;
T2:= NOT LT24;
T3:= TIMER_2.Q;
T4:= TIMER_1.Q AND (NOT LT44) AND LT24;
T5:= NOT LT24;
T6:= NOT LT44 OR NOT LT43;
T7:= TIMER_3.Q;

// ETAPES DEL GRAFCET
CASE ESTAT OF
60:
    IF (T0 = TRUE) THEN
        ESTAT:= 61;
    ELSIF (T4 = TRUE) THEN
        ESTAT:= 64;
    END_IF
61:
    IF (T1 = TRUE) THEN
        ESTAT:= 62;
    END_IF
62:
    IF (T2 = TRUE) THEN
        ESTAT:= 63;
    END_IF
63:
    IF (T3 = TRUE) THEN
        ESTAT:= 40;
    END_IF
64:
    IF (T5 = TRUE) THEN
        ESTAT:= 65;
    END_IF
```

```
65:      IF (T6 = TRUE) THEN
          ESTAT:= 66;
        END_IF
66:      IF (T7 = TRUE) THEN
          ESTAT:= 40;
        END_IF
    END_CASE

    // TEMPORITZADORS
    TIMER_1(IN:= (ESTAT = 60), PT:= T#1s);
    TIMER_2(IN:= (ESTAT = 63), PT:= T#1s);
    TIMER_3(IN:= (ESTAT = 66), PT:= T#1s);

    // SORTIDES
    IF (ESTAT = 60) THEN
        L_START:= FALSE;
    END_IF

    IF (ESTAT = 61) THEN
        V421:= TRUE;
        V442:= TRUE;
        PUMP:= TRUE;
        L_START:= TRUE;
    END_IF

    IF (ESTAT = 62) THEN
        V422:= TRUE;
        V441:= TRUE;
        V442:= FALSE;
        V421:= FALSE;
    END_IF

    IF (ESTAT = 63) THEN
        PUMP:= FALSE;
        V422:= FALSE;
        V441:= FALSE;
    END_IF












    IF (ESTAT = 64) THEN
        V422:= TRUE;
        V441:= TRUE;
        PUMP:= TRUE;
        L_START:= TRUE;
    END_IF

    IF (ESTAT = 65) THEN
        V422:= FALSE;
        V441:= FALSE;
        V442:= TRUE;
        V421:= TRUE;
    END_IF

    IF (ESTAT = 66) THEN
        PUMP:= FALSE;
        V421:= FALSE;
        V442:= FALSE;
    END_IF

END_PROGRAM
```

## 8.2.7 Programa Recepta 6

Name	Type
 T0	BOOL
 T1	BOOL
 T2	BOOL
 T3	BOOL
 T4	BOOL
 T5	BOOL
 T6	BOOL
 T7	BOOL
 TIMER_1	TON
 TIMER_2	TON
 TIMER_3	TON

PROGRAM \_INIT

```
V431:= FALSE;
V432:= FALSE;
V441:= FALSE;
V442:= FALSE;
PUMP:= FALSE;
L_START:= FALSE;
```

END\_PROGRAM

PROGRAM \_CYCLIC

```
// ACTUALITZACIÓ ENTRADES
T0:= TIMER_1.Q AND (NOT LT34) AND LT44;
T1:= NOT LT44 OR NOT LT43;
T2:= NOT LT34;
T3:= TIMER_2.Q;
T4:= TIMER_1.Q AND (NOT LT44) AND LT34;
T5:= NOT LT34;
T6:= NOT LT44 OR NOT LT43;
T7:= TIMER_3.Q;

// ETAPES DEL GRAFCET
CASE ESTAT OF
  70:
    IF (T0 = TRUE) THEN
      ESTAT:= 71;
    ELSIF (T4 = TRUE) THEN
      ESTAT:= 74;
    END_IF
  71:
    IF (T1 = TRUE) THEN
      ESTAT:= 72;
    END_IF
  72:
    IF (T2 = TRUE) THEN
      ESTAT:= 73;
    END_IF
  73:
    IF (T3 = TRUE) THEN
      ESTAT:= 40;
    END_IF
  74:
    IF (T5 = TRUE) THEN
      ESTAT:= 75;
    END_IF
```

```
75:      IF (T6 = TRUE) THEN
          ESTAT:= 76;
      END_IF
76:      IF (T7 = TRUE) THEN
          ESTAT:= 40;
      END_IF
END_CASE

// TEMPORITZADORS
TIMER_1(IN:= (ESTAT = 70), PT:= T#1s);
TIMER_2(IN:= (ESTAT = 73), PT:= T#1s);
TIMER_3(IN:= (ESTAT = 76), PT:= T#1s);

// SORTIDES
IF (ESTAT = 70) THEN
    L_START:= FALSE;
END_IF

IF (ESTAT = 71) THEN
    V431:= TRUE;
    V442:= TRUE;
    PUMP:= TRUE;
    L_START:= TRUE;
END_IF

IF (ESTAT = 72) THEN
    V432:= TRUE;
    V441:= TRUE;
    V442:= FALSE;
    V431:= FALSE;
END_IF

IF (ESTAT = 73) THEN
    PUMP:= FALSE;
    V432:= FALSE;
    V441:= FALSE;
END_IF

IF (ESTAT = 74) THEN
    V432:= TRUE;
    V441:= TRUE;
    PUMP:= TRUE;
    L_START:= TRUE;
END_IF

IF (ESTAT = 75) THEN
    V432:= FALSE;
    V441:= FALSE;
    V442:= TRUE;
    V431:= TRUE;
END_IF

IF (ESTAT = 76) THEN
    PUMP:= FALSE;
    V431:= FALSE;
    V442:= FALSE;
END_IF

END_PROGRAM
```



### 8.2.8 Programa control de visualització

Name	Type
◆ AUTO_MAN_TEXT	STRING[80]
◆ A_M_1_TEXT	STRING[80]
◆ A_M_2_TEXT	STRING[80]
◆ A_M_3_TEXT	STRING[80]
◆ INICIAR_PANTALLA_1	BOOL
◆ INICIAR_PANTALLA_2	BOOL
◆ INICIAR_PANTALLA_3	BOOL
◆ INTERMITENCIA_ALARM A_1	TON
◆ INTERMITENCIA_ALARM A_2	TON
◆ INTERMITENCIA_ALARM A_3	TON
◆ LT_11_ON_OFF	BOOL
◆ LT_12_ON_OFF	BOOL
◆ LT_13_ON_OFF	BOOL
◆ LT_14_ON_OFF	BOOL
◆ LT_15_ON_OFF	BOOL
◆ LT_16_REAL	REAL
◆ LT_21_ON_OFF	BOOL
◆ LT_22_ON_OFF	BOOL
◆ LT_23_ON_OFF	BOOL
◆ LT_24_ON_OFF	BOOL
◆ LT_25_ON_OFF	BOOL
◆ LT_26_REAL	REAL
◆ LT_31_ON_OFF	BOOL
◆ LT_32_ON_OFF	BOOL
◆ LT_33_ON_OFF	BOOL
◆ LT_34_ON_OFF	BOOL
◆ LT_35_ON_OFF	BOOL
◆ LT_36_REAL	REAL
◆ POW_1_ON_OFF	BOOL
◆ POW_2_ON_OFF	BOOL
◆ POW_3_ON_OFF	BOOL
◆ PT_11_REAL	REAL
◆ PT_21_REAL	REAL
◆ PT_31_REAL	REAL
◆ P_11_ACT	REAL
◆ P_11_ON_OFF	BOOL
◆ P_21_ACT	REAL
◆ P_21_ON_OFF	BOOL
◆ P_31_ACT	REAL
◆ P_31_ON_OFF	BOOL

◆ RESET_PANTALLA_1	BOOL
◆ RESET_PANTALLA_2	BOOL
◆ RESET_PANTALLA_3	BOOL
◆ SP_1	INT
◆ SP_2	INT
◆ SP_3	INT
◆ START_PANTALLA_1	BOOL
◆ START_PANTALLA_2	BOOL
◆ START_PANTALLA_3	BOOL
◆ STOP_PANTALLA_1	BOOL
◆ STOP_PANTALLA_2	BOOL
◆ STOP_PANTALLA_3	BOOL
◆ SV_1	INT
◆ SV_2	INT
◆ SV_3	INT
◆ V_11_ON_OFF	BOOL
◆ V_12_ON_OFF	BOOL
◆ V_13_ON_OFF	BOOL
◆ V_14_ON_OFF	BOOL
◆ V_15_ACT	REAL
◆ V_21_ON_OFF	BOOL
◆ V_22_ON_OFF	BOOL
◆ V_23_ON_OFF	BOOL
◆ V_24_ON_OFF	BOOL
◆ V_25_ACT	REAL
◆ V_31_ON_OFF	BOOL
◆ V_32_ON_OFF	BOOL
◆ V_33_ON_OFF	BOOL
◆ V_34_ON_OFF	BOOL
◆ V_35_ACT	REAL

PROGRAM \_CYCLIC

```
//ASSIGNACIÓ TEXT AUTOMÀTIC O MANUAL
IF (AUTO_MAN = 0) THEN
    AUTO_MAN_TEXT:= 'AUTOMATIC';
ELSE
    AUTO_MAN_TEXT:= 'MANUAL';
END_IF

IF (A_M_1_ESTAT = 0) THEN
    A_M_1_TEXT:= 'AUTOMATIC';
ELSE
    A_M_1_TEXT:= 'MANUAL';
END_IF

IF (A_M_2_ESTAT = 0) THEN
    A_M_2_TEXT:= 'AUTOMATIC';
ELSE
    A_M_2_TEXT:= 'MANUAL';
END_IF

IF (A_M_3_ESTAT = 0) THEN
    A_M_3_TEXT:= 'AUTOMATIC';
ELSE
    A_M_3_TEXT:= 'MANUAL';
END_IF
```

```
// ASSIGNACIÓ ESTATS ALS ACTUADORS DE LA PANTALLA MAQUETA 1
IF P_11 > 0 THEN
    P_11_ON_OFF:= TRUE;
ELSE
    P_11_ON_OFF:= FALSE;
END_IF

IF LT_11 = TRUE THEN
    LT_11_ON_OFF:= TRUE;
ELSE
    LT_11_ON_OFF:= FALSE;
END_IF

IF LT_12 = TRUE THEN
    LT_12_ON_OFF:= TRUE;
ELSE
    LT_12_ON_OFF:= FALSE;
END_IF

IF LT_13 = TRUE THEN
    LT_13_ON_OFF:= TRUE;
ELSE
    LT_13_ON_OFF:= FALSE;
END_IF

IF LT_14 = TRUE THEN
    LT_14_ON_OFF:= TRUE;
ELSE
    LT_14_ON_OFF:= FALSE;
END_IF

IF LT_15 = TRUE THEN
    LT_15_ON_OFF:= TRUE;
ELSE
    LT_15_ON_OFF:= FALSE;
END_IF

IF V_11 = TRUE THEN
    V_11_ON_OFF:= TRUE;
ELSE
    V_11_ON_OFF:= FALSE;
END_IF

IF V_12 = TRUE THEN
    V_12_ON_OFF:= TRUE;
ELSE
    V_12_ON_OFF:= FALSE;
END_IF

IF V_13 = TRUE THEN
    V_13_ON_OFF:= TRUE;
ELSE
    V_13_ON_OFF:= FALSE;
END_IF

IF V_14 = TRUE THEN
    V_14_ON_OFF:= TRUE;
ELSE
    V_14_ON_OFF:= FALSE;
END_IF

IF POWER_1 = TRUE THEN
    POW_1_ON_OFF:= TRUE;
ELSE
    POW_1_ON_OFF:= FALSE;
END_IF

IF START_PANTALLA_1 = TRUE THEN
```

```
        START_1 := TRUE;
ELSE
    START_1:= FALSE;
END_IF

IF STOP_PANTALLA_1 = TRUE THEN
    STOP_1 := TRUE;
ELSE
    STOP_1:= FALSE;
END_IF

IF RESET_PANTALLA_1 = TRUE THEN
    RESET_1 := TRUE;
ELSE
    RESET_1:= FALSE;
END_IF

IF INICIAR_PANTALLA_1 = TRUE THEN
    INICIAR_1 := TRUE;
ELSE
    INICIAR_1 := FALSE;
END_IF

P_11_ACT:= ((P_11 * 100) / 32767);
V_15_ACT:= ((V_15 * 100) / 32767);
PT_11_REAL:= ((PT_11 * 100) / 32767);
LT_16_REAL:= ((LT_16 * 100) / 32767);
SP_1:= CONSIGNA_1;
SV_1:= ((PID_OUT_1 * 100) / 32767);

// ASSIGNACIÓ ESTATS ALS ACTUADORS DE LA PANTALLA MAQUETA 2
IF P_21 > 0 THEN
    P_21_ON_OFF:= TRUE;
ELSE
    P_21_ON_OFF:= FALSE;
END_IF

IF LT_21 = TRUE THEN
    LT_21_ON_OFF:= TRUE;
ELSE
    LT_21_ON_OFF:= FALSE;
END_IF

IF LT_22 = TRUE THEN
    LT_22_ON_OFF:= TRUE;
ELSE
    LT_22_ON_OFF:= FALSE;
END_IF

IF LT_23 = TRUE THEN
    LT_23_ON_OFF:= TRUE;
ELSE
    LT_23_ON_OFF:= FALSE;
END_IF

IF LT_24 = TRUE THEN
    LT_24_ON_OFF:= TRUE;
ELSE
    LT_24_ON_OFF:= FALSE;
END_IF

IF LT_25 = TRUE THEN
    LT_25_ON_OFF:= TRUE;
ELSE
    LT_25_ON_OFF:= FALSE;
END_IF

IF V_21 = TRUE THEN
```

```
V_21_ON_OFF:= TRUE;
ELSE
    V_21_ON_OFF:= FALSE;
END_IF

IF V_22 = TRUE THEN
    V_22_ON_OFF:= TRUE;
ELSE
    V_22_ON_OFF:= FALSE;
END_IF

IF V_23 = TRUE THEN
    V_23_ON_OFF:= TRUE;
ELSE
    V_23_ON_OFF:= FALSE;
END_IF

IF V_24 = TRUE THEN
    V_24_ON_OFF:= TRUE;
ELSE
    V_24_ON_OFF:= FALSE;
END_IF

IF POWER_2 = TRUE THEN
    POW_2_ON_OFF:= TRUE;
ELSE
    POW_2_ON_OFF:= FALSE;
END_IF

IF START_PANTALLA_2 = TRUE THEN
    START_2 := TRUE;
ELSE
    START_2:= FALSE;
END_IF

IF STOP_PANTALLA_2 = TRUE THEN
    STOP_2 := TRUE;
ELSE
    STOP_2:= FALSE;
END_IF

IF RESET_PANTALLA_2 = TRUE THEN
    RESET_2 := TRUE;
ELSE
    RESET_2:= FALSE;
END_IF

IF INICIAR_PANTALLA_2 = TRUE THEN
    INICIAR_2 := TRUE;
ELSE
    INICIAR_2 := FALSE;
END_IF

P_21_ACT:= ((P_21 * 100) / 32767);
V_25_ACT:= ((V_25 * 100) / 32767);
PT_21_REAL:= ((PT_21 * 100) / 32767);
LT_26_REAL:= ((LT_26 * 100) / 32767);
SP_2:= CONSIGNA_2;
SV_2:= ((PID_OUT_2 * 100) / 32767);

// ASSIGNACIÓ ESTATS ALS ACTUADORS DE LA PANTALLA MAQUETA 3
IF P_31 > 0 THEN
    P_31_ON_OFF:= TRUE;
ELSE
```

```
P_31_ON_OFF:= FALSE;
END_IF

IF LT_31 = TRUE THEN
    LT_31_ON_OFF:= TRUE;
ELSE
    LT_31_ON_OFF:= FALSE;
END_IF

IF LT_32 = TRUE THEN
    LT_32_ON_OFF:= TRUE;
ELSE
    LT_32_ON_OFF:= FALSE;
END_IF

IF LT_33 = TRUE THEN
    LT_33_ON_OFF:= TRUE;
ELSE
    LT_33_ON_OFF:= FALSE;
END_IF

IF LT_34 = TRUE THEN
    LT_34_ON_OFF:= TRUE;
ELSE
    LT_34_ON_OFF:= FALSE;
END_IF

IF LT_35 = TRUE THEN
    LT_35_ON_OFF:= TRUE;
ELSE
    LT_35_ON_OFF:= FALSE;
END_IF

IF V_31 = TRUE THEN
    V_31_ON_OFF:= TRUE;
ELSE
    V_31_ON_OFF:= FALSE;
END_IF

IF V_32 = TRUE THEN
    V_32_ON_OFF:= TRUE;
ELSE
    V_32_ON_OFF:= FALSE;
END_IF

IF V_33 = TRUE THEN
    V_33_ON_OFF:= TRUE;
ELSE
    V_33_ON_OFF:= FALSE;
END_IF

IF V_34 = TRUE THEN
    V_34_ON_OFF:= TRUE;
ELSE
    V_34_ON_OFF:= FALSE;
END_IF

IF POWER_3 = TRUE THEN
    POW_3_ON_OFF:= TRUE;
ELSE
    POW_3_ON_OFF:= FALSE;
END_IF

IF START_PANTALLA_3 = TRUE THEN
    START_3 := TRUE;
ELSE
    START_3:= FALSE;
END_IF
```

```
IF STOP_PANTALLA_3 = TRUE THEN
    STOP_3 := TRUE;
ELSE
    STOP_3:= FALSE;
END_IF

IF RESET_PANTALLA_3 = TRUE THEN
    RESET_3 := TRUE;
ELSE
    RESET_3:= FALSE;
END_IF

IF INICIAR_PANTALLA_3 = TRUE THEN
    INICIAR_3 := TRUE;
ELSE
    INICIAR_3 := FALSE;
END_IF

P_31_ACT:= ((P_31 * 100) / 32767);
V_35_ACT:= ((V_35 * 100) / 32767);
PT_31_REAL:= ((PT_31 * 100) / 32767);
LT_36_REAL:= ((LT_36 * 100) / 32767);
SP_3:= CONSIGNA_3;
SV_3:= ((PID_OUT_3 * 100) / 32767);

END_PROGRAM
```

